

AMIGA MAGAZIN

Csak az Amigáról fehérén feketén



Hivatalosan szponzorálja:



FEJES
STUDIO

AMIGA MAGAZIN

Szia !

Itt van az Amiga Magazin harmadik száma is. Ezentúl minden hónap végén igyekszünk postosan megjeleníteni, hogy lehessen rá számítani. Sajnos bármikor közbejöhet valami, ami késedelmet okozhat. Ennek az elkerülése egyetlen módon lehetséges: sok pénzzel. Az Amiga Magazin pedig nem gazdag vállalkozás...

Ha már itt tartunk, szeretnék válaszolni egy nyájas olvasó kérdésére, tudniillik, hogy miért ilyen drága ez az újság. Tulajdonképpen azt hiszem válaszolhatnék úgy is, hogy akinek az Amiga Magazin nem ér meg 148 forintot, ami ma már nagyon kevés pénz, azt valószínűleg nem is érdekli az egész gép. Másrészt viszont a szokásos válasz: alacsony példányszám, magas nyomdaköltség, postai díjak, adó. Azt hiszem nem mi próbálunk meg nagyot harapni, hanem az átlagos életszínvonal tragikus az országban. Egyébként az lenne az ideális, ha mi csak az AM-mel foglalkozhatnánk, nem kellene más munkából megéljünk, és akkor sokkal jobb lehetne az újság.

Reméljük az újság egyre jobban tetszik mindenkinek. A kinézetét megpróbáljuk javítani, ami nagyon-nagyon fontos, de nagy változtatásnak nincs realitása ebben a pillanatban.

Az újság tartalmát megpróbáljuk egy kicsit színesíteni, több kézzelfogható, közvetlenül felhasználható információval. Szerencsére kapcsolatba kerültünk néhány emberkével, akiktől ilyen információkat kapunk is. Öröndetes tény, hogy kezdenek jönni számunkra az anyagok. Több közölhető írást kaptunk olvasóinktól, de jönnek a kérdések is. Nagyon örülünk mind a kettőnek.

Mindazonáltal szeretnénk sokkal több információt kapni: programleírásokat, amelyek még nem jelentek meg. Ez nagyon megkönnyítené a munkánkat, hiszen egy program leírásának elkészítése, de még lefordítása is hosszú időt vesz igénybe. Néhányan már küldtek leírásokat, például az F18 vadásszimulátorét, aminek angolul nekünk is megvolt, de így megmenekültünk a fordítás kínjától. Kaptunk továbbá felajánlást a Public Domain könyvtárunk bővítése céljából, amit szintén nagyon köszönünk ! Nem is hinnétek mekkora hiányt szenvedünk játékprogram ismertetőkből. Sajnos közöttünk nincs egy "játékörült" sem, így nagyon szívesen veszünk ilyen leírásokat is. Nem csak az újakat, lehet régi is, ha jó ! Ilyenkor szükségünk van a programra (ha több lemezes, elég annyi, amiből fotókat lehet készíteni), és egy rövidebb, vagy hosszabb leírásra (amennyit érdemel).

Minden levelet természetesen elolvassuk. Sajnos néha nagyon soká, illetve egyáltalán nem is válaszolunk rájuk, ha az nem feltétlenül indokolt. Reméljük ennek az okát megértitek.

Sajnos URZ totális megbízhatatlansága miatt vissza kellett állnunk az AMIGA DTP-re. Ha ez újbóli minőségromlást eredményezne az újság kinézetében, kérünk benneteket, tanúsítsatok megértést, és nyújtson számunkra kárpótlást, hogy az AM-nek nem kell szégyenszemre ATARI-n készülni. Jelen pillanatban még előtte állunk a januári számban leírt DTP 7.14 MHz-enek, az idegállapotunk aránylag normálisnak mondható.

Nagy segítséget nyújt számunkra az ANUBIS Kft., akik rendelkezésünkre bocsátottak egy hardisk-et, ami nagy mértékben megkönnyíti a munkánkat. A lézernyomtatást is náluk végezzük.

Elnézést kell kérnünk mindenkitől, aki lemezt rendelt meg tőlünk. A decemberi és januári ON DISK-eket február elején postáztuk csak ki, a februári számhoz tartozó lemezt és a Public Domain lemezeinket pedig a februári számmal együtt, időben küldjük. Reméljük a jövőben nem fog ilyen késlekedés előfordulni.

Köszönettel:

BB KING

Kavalkád: BEMUTATKOZÁS

Kosir Attila (THE MAD)

Első rossz húzás:

1971-ben születtem. Délelőtti esti gimnáziumba járok. Plus négyen kezdtem el normális programokat írni. Nyertem grafikai pályázatot, Plus 4-re írtam 4 játékprogramot, amiket el is adtam. (Digital Ball, Godzilla, Mesél az erdő, Invincible). Jelenleg Amigával foglalkozom. (Játékokat írok rá...) Főbb területeim a játék kódolása, és a grafikák elkészítése. Időnként DPAINT-ban cruncher-t kell, hogy írjak.

A második rossz húzás:

Az AM-ben tippeket fogok írni azoknak, akik játékprogramokat kívánnak fejleszteni.

Harmadik rossz húzás:

Ha valaki játékot ír, és grafikára van szüksége, szívesen segítek (az anyagi részleteket telefonon megbeszélhetjük). Pályákat, szörnyetegeket, úrhajókat, BRONX formákat, stb. rutinból rajzolok, forduljon hozzám bizalommal!

Az egyetlen jó húzás:

Az én rovatom SEKA felségterület. Több barátom azért átkozta el az AM-et, mert úgy lehúzták az egyetlen normális programot az Amigán, a SEKA-t. (Nem vagyok én macska, hogy a DEVPAC-ban játszadok az egérrel...)

Kovács Zsolt

1965-ben születtem Budapesten. Bölcsészkaron végeztem magyar-angol szakon, ennek ellenére mindig is foglalkoztam számítástechnikával, és gyerekkorom óta vonzott az elektronikus zene. A jó öreg C64-re számos midi-programot gyártottam, főleg házi használatra, azután átnyergeltem az Amigára. Jelenleg filmek fordításából élek (de kizárólag feliratozás és szinkron céljából, mivel viszolygok a lélektelen és amatőr alámondásoktól!!!), valamint reklámfilmekhez gyártok zenéket. Az elektronikus hangszerek igen nagy ismerőjének tartom magam, és efféle műveltségemet igyekszem majd propagálni az Amiga magazin hasábjain is. Mivel időnként kissé idealista vagyok, hiszek abban, hogy elszánt erőfeszítésekkel végül is lehet apró eredményeket elérni a hazai film- és videós világban (és országunkban sajtólatos módon eléggé általánosan) tomboló látványos amatőrizmus öntelt réme ellen.

Varga Csaba,

1964 február 24. (okl. prog-mat) - (Aniware)

1985 januárban a kollégium C-64-esével kezdtem a számítástechnikát. Nemi basic után rögtön assembler. Először csak néhány játékban egy-két ellenséges figurát tettem láthatatlanná, de élők maradtak. Így a jelentősen megnehezített játékok miatt hamar "népszerűséget" vívtam ki magamnak. Noha szándékos rosszindulatból soha nem csináltam semmit, utánam mégis mindig kikapcsolták a gépet 1 percre.

Amigám 1988 nyaratól van, a Mariahilfer str -i egyik Herlango-ból származik. Irodalom híján fél évig csak a Workbench 1.2 és Extras D1.2 bámulatos képességeiben gyönyörködhettem. Lassan azért sikerült néhány fontosat megszerezni. ROM-lista néhány részlete, függvényleírások, software-leírások (C, ass.), gyári kapcsolási rajz, belső felépítés, 68000 könyv és ...

HW Elektronikával 12 éve foglalkozom. Analóg és digitális egyaránt. Noha intézményesen soha sehol nem tanultam, kizárólag otthon, azért nem panaszkodhatom. Néhány saját ötleten alapuló, saját tervezésű és a gyakorlatban is kipróbált áramköröm cikk-ként megjelent a Commodore Újságban, Magyar elektronikában és Rádiótechnikában. Szenzoros botkormány, C-64 eprom égető, speciális memóriabővítő, amellyel felülről 100% kompatibilis marad az alapgép, normál 4-külön kurzor billentyű C-64-re hardware-ből (szintén 100% kompatibilitású) és sok más, újságban nem megjelent dolog.

Beleznay Annamária



A New York-i Parsons Főiskolán szerezte diplomáját, és tizenöt éve dolgozik a reklámgrafika területén Görögországban és Magyarországon. A kalapos emberke az ő találmánya, copyright védi.

Ő ad nekünk tanácsokat az AM kinézetének a javításához, ami bőven ránk is fér. Reméljük a jövőben is sok tanácsot kapunk tőle, és az újság egyre szebb lesz...

Kavalkád: JÁTÉK

WARHEAD

Ez a szöveg a WARHEAD nevű játék bejelentkezésekor íródik ki. Ha nem tudsz jól angolul, akkor ezt nem érted meg, és így a játék lényegét sem érted meg, ami pedig hiba lenne...

A DOKUMENTUM

Ez a dokumentum azoknak a pilótáknak az eligazítására készült akiknek ez az első szolgálatuk.

AZ ELLENSÉG

A konfliktus legzavaróbb oldala az, hogy nem ismerjük az ellenségünket. Nem tudjuk, hogy miért támadtak meg minket és azt sem, hogy kik egyáltalán, viszont tudjuk, hogy az otthonuk a Sirius csillag egyik bolygóján van és hogy, hogy néznek ki.

Az egyes lények formailag nagyon hasonlóak a földi rovarokra, habár nagyobbak és egy kicsit nehezebbek. Amikor a maradványaikat megtalálták, svábbogaraknak nevezték el őket. Ez a név rájuk ragadt.

A többi információnk róluk főleg feltevés. Ezek összefoglalva:

Azt gondoljuk, hogy ezeknek a lényeknek és a földi rovaroknak közös ősök lehet. Az alakjuk a bolygójukon uralkodó kellemetlen feltételeknek köszönhető. Ez az alak a technika fejlesztő és használó képességeik kárára van. Ha ez nem így lenne, már az egész Galaktikát elárasztották volna. A lények földi mércével nézve ősinek számítanak. Okunk van feltételezni, hogy a lények jelenlegi evolúciós állapotukat akkor érték el, amikor a Földön még a dinoszauruszok voltak a legmagasabb szintű élőlények. Úgyanúgy mint a földi rovarok az egyedek meglehetősen buták és tehetetlenek. Úgy működnek mint egy nagy közösség, amit a királynő irányít. Viszont nem kémiai anyagokkal (szagokkal, ízekkel) kommunikálnak egymással mint a földi rovarok. A "svábbogarak" teste vasrészecskéket tartalmaz, amik egy szerves adó és vevő készülékként viselkednek (telepatikusak). Ez az a képességük, amelyik nem csak az egymás közötti kapcsolatteremtést teszi lehetővé, hanem a korlátozott értelmük összekapcsolását egy csoportértelemmé. Megosztott öntudatuk van. Több tíz vagy még száz rovar is szükséges mielőtt bármilyen intelligens megnyilvánulást mutatnak. Vitatható hogy a szíriuszi-szövetség egyetlen egy hatalmas élőlény, vagy pedig egyedekből áll.

Úgy néz ki, hogy a műszaki eszközök gyártása meglehetősen új ezeknek a rovarszerű lényeknek. Úgy gondoljuk, hogy az űrhajógyártási képességeik korlátozottak. Nem tudjuk hogy mi motiválta ezeket a lényeket a űr meghódítására. Nekünk egyszerűen szembe kellett néznünk a döntésük szörnyű következményeivel. Az első szíriuszi űrhajót 1897-ben lőtték ki a Szíriusz-Ötről. Azóta átkutatták a különböző csillagrendszereket ebben a spirálkarban. Elképzelhetetlen, hogy az eltelt időben nem látogatták meg a Földet.

A KONFLIKTUS

2045-ben a lények újra meglátogatták bolygónkat. Egy óriási iszonyú fegyverekkel felszerelt űr flotta érkezett a mit sem sejtő Föld megtámadására. Több millió tonna fegyvert vetettek be. A Földet teljesen letarolták. Magában a csatában több mint 3 milliárd ember vesztette életét. A következő években a népesség több mint egyharmada halt meg a konfliktus következményei miatt. Úgy gondoljuk, hogy ennek a támadásnak a célja a Föld népeinek barbarizmusba való lealacsonyítása volt. A rovarok alábecsülték a felépülési képességeinket, különben már visszajöttek volna. Valószínűnek tartjuk, hogy egy nap visszatérnek, hogy befejezzék a munkát amit 12 évvel ezelőtt elkezdtek.

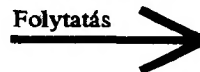
POLITIKA

Az elmúlt 12 év történelme figyelemre méltó. A nemzetek közötti megosztottság teljesen megszűnt. A Fist-Of-Earth szervezet egy rendkívül jó hatékonyságú hadi kormányt hozott létre. A teljesen lepusztult Föld helyreállításának megkezdése mellett, a Fist-Of-Earth kormány hozzáfogott egy védekező erő kialakításához, ami a Földet óvna a további támadásoktól. A védelmi politika élén a Napbázis Földkörüli pályán való megalkotása szerepelt.

A NAPBÁZIS

Ez az Űrerőd most az Új Védelmi Kezdeményezés központja. Alapvető szerepet játszik az emberiség önmaga megvédésére végzett próbálkozásaiban. Miután a Bázis üzemelni kezdett, Napkörüli pályára állították.

Folytatás



Kavalkád: JÁTÉK

PD

Egy olyan pályára amelyik a Naprendszer lehető legnagyobb védelmi lehetőségét kínálja. A bázist a Földről induló szállítóűrhajók látják el alkatrészekkel, legénységgel és utánpótlással. A bázis kereszt alakú és körbe forog, hogy gravitációt hozzon létre. A kereszt középpontja dokkolási lehetőséget nyújt a beérkező űrhajóknak. A hajókat egy lifttel lehet 'levinni' a kilövő kamrába. A kamrában levő űrhajókat ki tudjuk 'dobni' (lőni) az űrbe. A bázis másik három karja a hálókabin egységeket és a parancsnoki főhadiszállást foglalja magában. A Napbázis kétirányú mikro-rádióadások kibocsátására képes, így néhány másodperc alatt bármelyik közelebbi űrhajóval kapcsolatot tud teremteni.

Úgy döntöttünk, hogy visszavágnunk és megvédjük magunkat. Végre van egy fegyverünk: a FOE-57 űrhajó.

AZ ŰRHAJÓ

A FOE-57 az első működőképes csillagközi vadász-űrrepülőgép. A hajó képes keresztül haladni a magasabb dimenziókon. Ezt a technológiát használva az űrhajó bármilyen közelebbi naprendszerbeli pontot gyorsan el tud érni. A FOE-57 nagy tömegű fegyvereket szállíthat, amiket gyorsan és pontosan tud alkalmazni.

Ez az az űrhajó amit a szíriuszi-szövetség erői ellen fogsz használni.

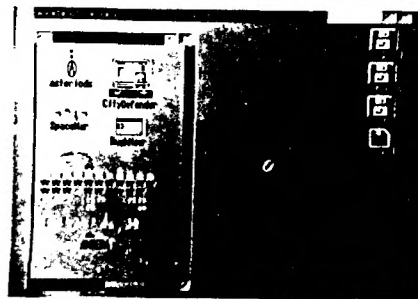
A Föld népeinek biztonsága a kezében van!

A-Szöveg-Vége

(Fordította: ALEX)

**Csatlakozz
Hozzánk!**

Küldj nekünk anyagot! Várunk új, és régi, de nem ismert programok leírását. Ilyenkor kérünk egy lemezt is, ami alapján fényképet készíthetünk! Lehet játék, és felhasználói program is. Néhány már érkezett, amit nagyon köszönünk!



Sokan a hasukat fogják a röhögéstől, ha PUBLIC DOMAIN játékokról esik szó. Ezek általában nem a csodálatosan megrajzolt grafikájukról, vagy káprázatos hanghatásokról híresek. Gyakran BASIC-ben írják őket, ami már önmagában kacagásra bírja a "nagyfejű" programozókat. Én azonban, még ha meg is köveznek érte: ezeket a játékokat élvezem a legjobban. Valahogy - talán mert nem is várok tőlük sokat - sokkal jobban tudok szórakozni velük. Egy példa: a jó öreg HT1080Z iskolaszámítógépen van egy Galaxy nevű játék. Egyike azon kevés HT játékoknak, amit Assembly-ben kódoltak. Aki nem ismerné a HT-t annak úgy tudnám röviden jellemezni, hogy belső hangszórójából csippanásszerű hangokat képes maximum kihozni, és a grafikája fekete-fehér! Egy pixel Kb. akkora, mint egy betű, amit most olvasol. Ezek után gondolom el tudod képzelni milyen volt ez a játék kinézete. A Gimnáziumban mégis olyan botrányok fordultak elő, hogy a gondosan kulcsra zárt technika teremében elbújtak az elvetemült diákok, és miután az utolsó takarító is elment, egészen másnap reggelig játszottak.

A Public Domain játékok általában egy nagyon jól játszható, egyszerű ötleten alapulnak. Ilyen például a TRON, amit a decemberi számunkban közöltünk. (A programlistát az Amiga Basic-be kell begépelni, és RUN-nal futtatni. Elindítható változatban rajta van a decemberi ON DISK lemezünkön) Ezt a programot egyébként már majdnem minden programozó megírta, akivel találkoztam. Szívesen vennék ha küldenének még néhány verziót! (küldhettek más saját készítésű játékot is)

A játék végtelenül egyszerű: a két játékosnak egy-egy vonalat kell irányítania. Aki beleütközik a másikba, vagy a képernyő szélébe az veszett. Természetesen az ötletet a végtelenségig lehet variálni (gyorsítás, a képernyőnél nagyobb pálya, átugrási lehetőség, lövöldözés, stb.)

A jövőben szeretnénk minél több ilyen PD programot bemutatni nektek, higgyétek el, ha az előítéleteiteket legyőzitek hatalmas szórakozás! Jelenleg rendelkezésünkre áll néhány játék, de előbb-utóbb kifogyunk. Küldjétek!

Kavalkád: JÁTÉK

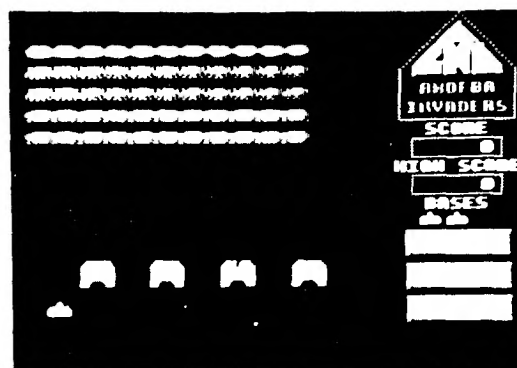
PD

Ezt a hírdetést ingyen emeltük ki, mint közérdekű információt. Biztosak vagyunk benne, hogy lesz erre a feladatra vállalkozó. Mi a DBase Amiga verzióját ajánljuk, ami elérhető Magyarországon. Elérhető továbbá a Superbase (ha valaki tudja használni, kérjük küldjön róla leírást is). Raktárkészletre kitűnően használható a Softwood File nevű program (GO AMIGO). Az igazi persze, ha valaki megírja a konkrét célt megvalósító programot...

Vásárolnék Amiga 500-hoz (1MB) Könyvelői, raktár- készletkezelő programokat. Esetleg elkészítését megrendelném.

Krasznai András

Tel.: 167-1439



JÖN !

A következő programok 1990 februárjában jelentették be hivatalosan USA-beli terjesztésre. Azért közöljük, hogy tudd, mi a "HOT STUFF"....:" (ezek közül néhan Európában már kapható)

ACCOLADE: Elvira - Mistress of the Dark - Stratego - Search for the king - Altered Destiny - Jack Nicklaus Unlimited Golf

DATA EAST: ABC's monday, Night football, The Dream Team, Full metal planet

DISNEY - dick tracy

ELECTRONIC ARTS: magic fly - Chuck Yeager's advanced flight trainer 2.0 - Block Racer - AD&D Pool of radiance (SSI) - TV SPORTS FOOTBALL II (Cinemaware) - Dragon Lord - Arcade Fever - Nightbreed (ocean) - 'N pile (UBISOFT)

ELECTRONIC ZOO - Berlin 1948, Treasure Trap

LIVE STUDIOS - Thunder Strike

KARMASOFT - Power Pinball

MASTERTRONIC - Super Off Road - Clue - Overlord

PSYGNOSIS - Matrix Marauders - Anarchy - Lemmings - Obitus -

Aquaventura - The Keep - Fire Stone -

Turbo Buggiers - Carthage - Planet

Busters - Barbarian II - Gore -Awesome

- Puggsy - Nitro

THREE-SIXTY - Harpoon - Mega Fortress - The Blue Max

Galaxy

A Galaxy az a játék, amiről a HT-s kaland esetében írtam az előző oldalon. A játékautomatákon mindenki által jól ismert Amigás változata. Természetesen ennél a programnál elképzelhető szebb kivitelezés, de ilyesmit Public Domain programnál nem illendő nézni (ajándék lónak ne nézd a fogát !) Ha ki akarsz próbálni, rajta van az ON DISK lemezünkön !

AM TOP 10 - játék

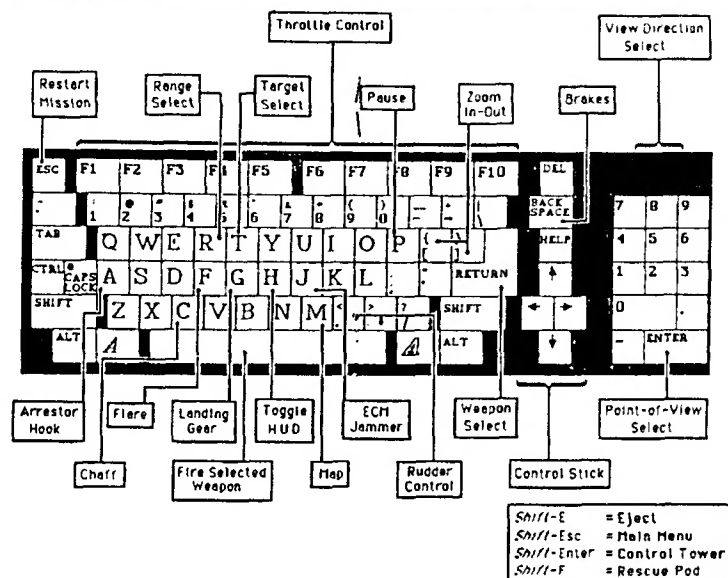
- 1 Cadaver
- 2 Powermonger
- 3 Chaos strikes back
- 4 Awesome
- 5 Future Wars
- 6 Lemmings
- 7 Pirates !
- 8 Wings
- 9 Supremacy
- 10 Team Yankee

AM TOP 10 - felhasználói

- 1 Deluxe Paint3.25
- 2 Diskmaster V1.3
- 3 Real 3D
- 4 Startrekker
- 5 Sculpt 4D
- 6 Deluxe Video III
- 7 Pagestream
- 8 Red Sector Demomaker
- 9 Superbase Professional
- 10 CygnusEd professional

Kavalkád: JÁTÉK

F/A 18



Kéri László:

Hogy senki ne mondhasa, hogy semmit sem tettem az AM-mért, íme az F18 Interceptor leírása:

AM:

Köszönjük!

F/A-18 INTERCEPTOR

Se szeri se száma az USA gépeinek szimulációs feldolgozásainak. Az egyik közülük az F/A-18 Interceptor, amelyben, mint nevéből is kiderül, egy F18-as vadászgépet kell irányítanunk San Francisco környékén. A játékban csak légi célpontokkal találkozunk.

(Az F18 Interceptor az egyik legjobb szimulációs program az Amigán, ami a repülés valóságsszerűségét illeti - AM)

A kezdet

Ha karriert akarunk játszani, a címképernyő után rakjunk be egy formátalt disket és nyomjuk le a RETURN-t. Ha volt valamilyen állás a disken, akkor a program azt betölti. Ellenkező esetben meg kell adni a nevünket és ezzel új karriert kezdünk.

A főmenü

A menüpontokat az 1-8 számbillentyűkkel választhatjuk ki.

- 1 - DEMO: demo
- 2 - FREE FLIGHT: szabad repülés, nincsenek ellenfelek. Először a kezdő repülőterert kell

megadni (1-4, 4 a repülőgépanyahajó), majd a gépünk típusát (1 F18, 2 F16). Ezután arra repülhetünk amerre akarunk (pl. a Golden Gate híd alá).

3 - TRAINING DEMO OF MANEUVERS: a program bemutatja a légi manővereket

4 - TRAINING PRACTICE OF MANEUVERS: harci manőverek gyakorlása. Itt először fel kell szállnunk úgy, hogy a motorteljesítmény 80% legyen majd utánaoznunk kell az előttünk haladó oktatógép manővereit. Ha elvégeztünk egy manővert, a tűzgomb megnyomásával kezdhetünk egy következőbe. Az aktuális manőver neve az üzenetsorban található.

Kilépés ebből a módból: SHIFT+ESC

Újrapozicionálás az oktatógép után: RETURN
Még egyszer RETURN: új manőver

Oktatógép kondenzcsíkot húz: S

5 - QUALIFICATION: minősítés. Hogy bevetésekre repülhessünk bizonyítanunk kell rátermettségünket ezzel a menüponttal. A feladat egyszerű: fel kell szállni az ENTERPRISE repülőgép hordozóról, majd le kell szállni rá.

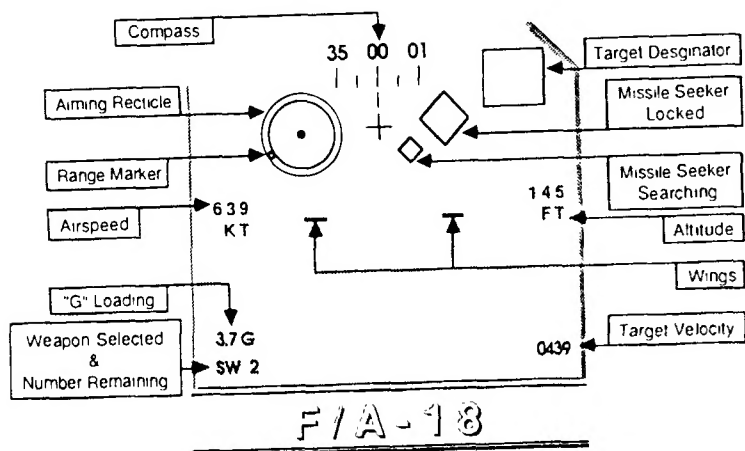
6 - SELECTABLE MISSIONS: belekezdés a választható bevetések valamelyikébe a megfelelő funkcióvillentyű megnyomásával.

Visual confirmation - idegen repülőgép azonosítása. Meg kell keresni az ellenséges gépet, azonosítani, majd visszarepülni a bázisra. Ha nem lönek rád, ne nyiss tüzet. Itt még választhatunk F16 és F18 között.

Emergency defence operation -Ellenséges gépek támadták meg az elnök gépét, ezért kényszerleszállást kell végrehajtania. A feladat az ellenséges gépek megsemmisítése. Egy üzenet tájékoztat, ha az elnöki gép egy darabban landolt, ilyenkor a feladatod "csak" annyi, hogy

Kavalkád: JÁTÉK

F/A 18



épségben visszakerülj a bázisra.

Intercept stolen aircraft - Két F-16-os Falcon átpártolt az ellenség oldalára, és éppen a Szovjetunió felé tartanak. Két MIG is kíséri őket. Meg kell őket találnod, és visszatérésre kényszeríteni őket. Először "békés" módon, de ha nem akarnak visszatérni szépszóval... A F16-osok nem szoktak visszatérni még a két MIG kilövése után sem.

Search and rescue operation - egy pilótátársadat lelőtte az ellenség, és katapultált. A radar és a "pilóta szemed" segítségével keresd meg szegényt. A térképen egyébként jelölve van a katapultálás helye, nem nehéz megtalálni, de hogy, hogy kell a csomagot ledobni, azt teljes homály fedi! (ha valaki a szerkesztőségéből tudja, írja le legyen szíves)
- SHIFT-F!

Ha egy bevetést sikeresen végrehajtottunk akkor megjelenik egy üzenet amiben gratulálnak, és megadják, hogy hol kell leszállnunk.

7 - NEXT ACTIVE ADVANCED MISSION: a következő küldetés indítása.

8 - YOUR CURRENT FLIGHT LOG STATISTICS: az eddigi eredményeink kijelzése. Ezek jelentése a következő:

Total missions started: az eddig megkezdett küldetések száma

Missions succesfully completed: a sikeresen végrehajtott küldetések száma

M61 cannon rounds: az eddig kilőtt gépágyú töltények száma (FIRED) és a találatok száma (HITS)

AIM-120 AMRAAMS: összesen kilőtt AMRAAM rakéták száma, és a találatok száma

AIM-9L sidewinders: összesen kilőtt

sidewinderek száma, és a találatok száma

Accum.flight time: az összesen repült idő

crashes: a lezuhanások száma

Hits from missiles: minket eltalált rakéták száma.

AZ ESC-el a főmenüben térhetünk vissza.

1-gyel a lemezre menthetjük az állást.

SHIFT+2-vel törölhetjük az eredményeket. Ha törölt állást mentünk a lemezre, újra megadhatjuk a nevünket.

Megjegyzendő, hogy a program biztonsági kódokat kérdez a repülések megkezdése előtt. A programhoz mellékelnek egy karton korongot, amin forgatható tárcsák vannak. Ez a "flight-computer". Ha a program biztonsági kódot kérdez, ezzel a tárcsával kell megkeresned a választ. (Persze ha valakinek feltört programja lenne, annak ezzel nem kell fáradnia).

A Fegyverzet:

Minden bevetésre ugyanazzal a fegyverzettel indulunk:

M61 Cannon: 20 mm-es gépágyú. Használata gyakorlatot igényel.

A M R A A M r a k é t a : k ö z e p e s hatótávolságú radarirányítású rakéta.

SIDEWINDER: rövid hatótávolságú infravörös önirányítású rakéta. Ebből csak kettő áll a rendelkezésünkre

A műszerfal:

- compass - iránytű
- aiming recticle - célzó
- range marker - távolság mérő
- altitude - magasság lábban
- "G" loading - nehézkedés. Ha ez az érték nagy, egy időre elsötétedik a képernyő (elájulunk).
- weapon selected & nr. remaining - az aktuális fegyver neve; és a rendelkezésünkre álló mennyiség
- target designator - célpont jelző
- missile seeker - célpont kereső célon
- missile seeker searching - célpont kereső keres
- altitude - magasság lábban
- wings - a szárnyak
- target velocity - a célpont sebessége

- brake indicator - fékviesszajelző
- ejection warning - katapultálási riadó
- gear - futómű, és fékezőhorog visszajelző
- ordinance display - fegyverzet mutató
- fuel indicator - üzemanyag mutató
- radar display - radar
- message window - üzenet ablak
- compass - iránytű

W - nyugat
E - kelet
N - észak
S - dél

- artificial horizon - műhorizont
- status display - állapot kijelző
- enemy near warning light - ellenséges gép a közelben lámpa
- friendly near warning light - baráti gép a közelben lámpa
- stall warning light - hajtóműhiba vissza jelző
- IR missile warning light - IR rakéta jelző
- RH missile warning light - RH rakéta jelző
- map position - a térkép helyzete

A billentyűzet

- restart mission az ablakon lévő kijelző BE/KI újratekésztése
- arrestor hook - fékező horog fel/le
- chaff - rakéták elleni küttyű
- flare - rakéták elleni küttyű
- range select - radar hatótávolság beállítása
- landing gear - futómű fel/le
- target select - célpont kiválasztás
- landing gear - futómű fel/le
- fire selected weapon - a kiválasztott fegyver kilövése
- toggle H.U.D - az sisakban lévő kijelző be/ki
- map - térkép
- throttle controll - üzemanyag
- pause - szünet
- ECM jammer - ellenséges radar butítása
- rudder controll - csúfítás
- weapon select - fegyverkiválasztás
- zoom in-out - ki-be közelítés a térképen
- control stick - botkormány
- brakes - fékek
- view direction select - kilátás iránya
- point of view - nézőpont (belső-külső)
- SHIFT-E - katapultálás
- SHIFT-ESC - főmenü
- SHIFT-ENTER - irányítótorny
- SHIFT-F - segélyszállítmány kidobása

Néhány instrukció:

A felszálláshoz adjunk 100%-os motorteljesítményt, és amikor a sebesség elért a 160 kts-t, felhúzzuk a gépet. Minél előbb húzzuk be a futóművet, mert sebességcsökkenést okoz. Felszállás után forduljunk az ellenséges gépek felé. Ha a radaron nem látszanak, a térképen is megkereshetjük őket. Ha már látszanak a radaron a "T" vel válasszunk célpontot. Az aktuális célpont adatai az üzenet ablakban látszanak (ALT magassága, HDG iránya, és SPD sebessége). Amikor először közelítjük meg az ellenfelet AMRAAM rakétát használunk, sidewindert ha közel van, mivel ezzel biztosabb a találat (kár, hogy csak kettő van belőle). Ha AMRAAM rakétát használunk próbáljuk meg befogásban tartani az ellenséget, így biztosabb a találat.

A leszálláshoz közelítsük meg a kívánt repteret, úgy, hogy a motorteljesítmény kb. 50% legyen, és a magasság sem legyen 200 láb felett. Próbáljunk meg úgy manőverezni, hogy a kifutópont szemben legyen velünk. Miután leraktuk a gépet egyből kapcsoljuk be a féket, hogy legyen helyünk az esetleges felszálláshoz. Ha az anyahajóra akarunk rászállni, akkor délről (nulla fok felől) kell megközelíteni a kifutót. Itt a futómű mellett, még a fékezőhorognak is kint kell lennie. A gépet még lehetőleg a fékezőhuzal előtt rakjuk le, hogy a horog biztosan megakadjon. Ilyenkor nem kell a féket bekapcsolni, a horog megfogja a gépet. Hogy újra elindulhassunk, húzzuk be a horgot.

Kéri László
Budapest
Amfiteátrum u. 26
1031

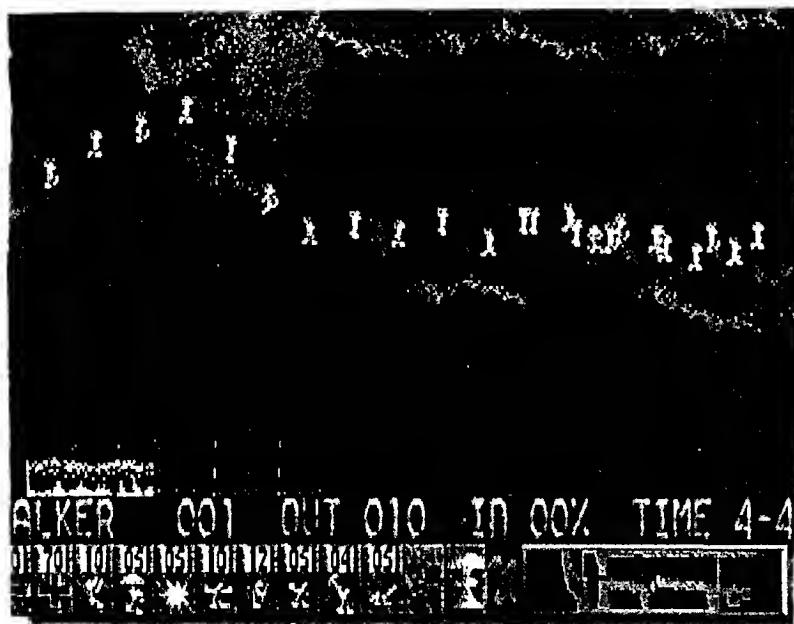
20.000 DM a legjobbnak !!!

A német Markt & Technik Verlag nem akármilyen pályázatot hirdetett meg! Bármilyen játékprogram beküldhető. Mellékelni kell a forráslistát és a dokumentációt lemezen és kinyomtatva. A program nem lehet üzleti forgalomban lévő. 1. díj 20.000 DM, 2. 6000 DM, 3. 2000 DM. A pályázat beadási határideje 1991. Április 15. A programmal küldeni egy kupont is, ami megtalálható a német Amiga Magazin januári számában. Ha van egy programod, amivel pályázni lehetne, amiben csak tudunk segíteni. Egy kuponunk van, azt a rendelkezésedre bocsátjuk, ha nem tudsz szerezni. Sok szerencsét kívánunk. Ha valaki megnyerné a "money"-t meghívhat bennünket egy STORMBRINGER-re (turbókártya)...



Kavalkád: JÁTÉK

A PSYGNOSIS ontja a jobbnál jobb programokat. Sokak véleménye szerint a játékprogramipar kétségtelenül legkimagaslóbb cége. Az angliai szürke városban, egy liverpool-i volt raktárépületben dolgozik 18 Amiga 300, és sok tucat 2000-esen a világ talán legjobb játékprogramgyártó team-je.



Lemmings

Egy fantasztikusan eredeti akciójáték ötlet. Nagyban különbözik a Psygnosis eddigi játékeitől, nemsoká piacra kerül.

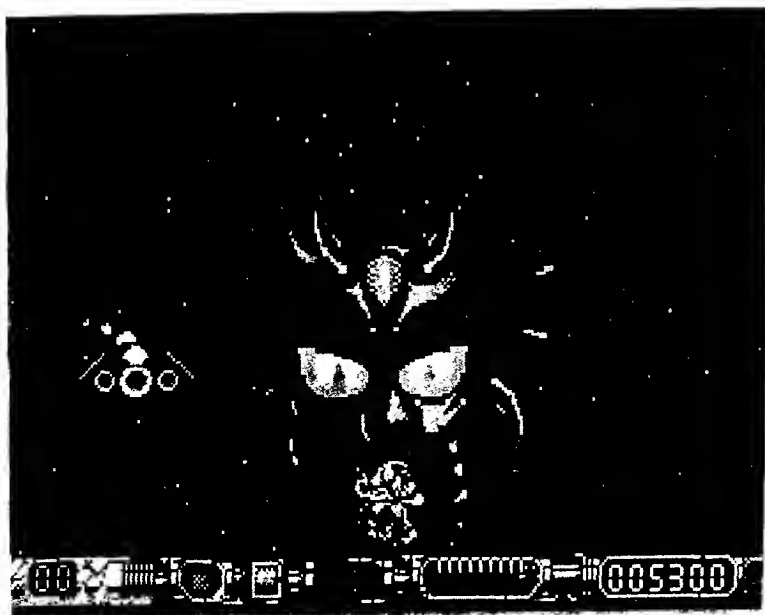
A lemmingeket (hogy azok mik, azt nem sikerült kiderítenünk...) meg kell mentenünk az öngyilkosságtól. A lemmingeknek alig valamivel alacsonyabb az öngyilkossági rátájuk a magyaroknál. Mindúntalan leugranak a sziklákról, de ha erre nincs lehetőségük, akkor is roppant találékonyak és valamit mindig kitalálnak. A játékos feladata, hogy bizonyos ikonokra klikkelgetve megmentse a lemmingeket az öngyilkosságtól.

Roppant szórakoztató játék, és tele van humorral. A grafika nem rossz, és a hanghatások pedig tökéletesek (elsősorban a lemming szétcsattanásának a hangja nagyon élethű, ha leesik a szikláról).

Awesome

Ha néhány hónappal előbb elkészül, valószínűleg ez a játék lett volna az év játékprogramja. Sebj, így is a PSYGNOSIS nyert. Ugyanaz a team fejlesztette, mint a Ballistix-et, és a BEAST I, II-t. A játék során több szinten kell keresztül menni. A különleges az az, hogy a nézőpont megváltozik mindegyiknél. Az első kettő szint egy egyszerű több irányba scrollozós pálya, ahol sokmindent szét kell durrantani... A harmadik egy nagyon hatásos rész. Az űrhajód megdől, és begyorsul a hipertérbe. Le kell győzöd a sárkányt, aki a bolygót védi, ahova le kell szállnod. A következő egy fentről nézett látkép, ahol a földalatti részbe kell lejuss, hogy elvégezd a feladatod...

Az Awesome egy hatalmas játék, amit nagyon élvezetessé tettek, hiszen az első két szint viszonylag egyszerű, nem ér folyton kudarc. Az animáció, a grafika, és a zene pedig Psygnosis szintű...



Kavalkád: JÁTÉK

Psygnosis

SPELLBOUND

Ez a játék tulajdonképpen konvencionálisnak tekinthető, hiszen liftekkel, különböző szinteken kell mozogni, ami elég hagyományos. Két játékos játszhatja, és a 8 pályán való túljutáshoz, a teljes győzelemhez jó együttműködésre van szükség. A grafika szép, a zene szuper.

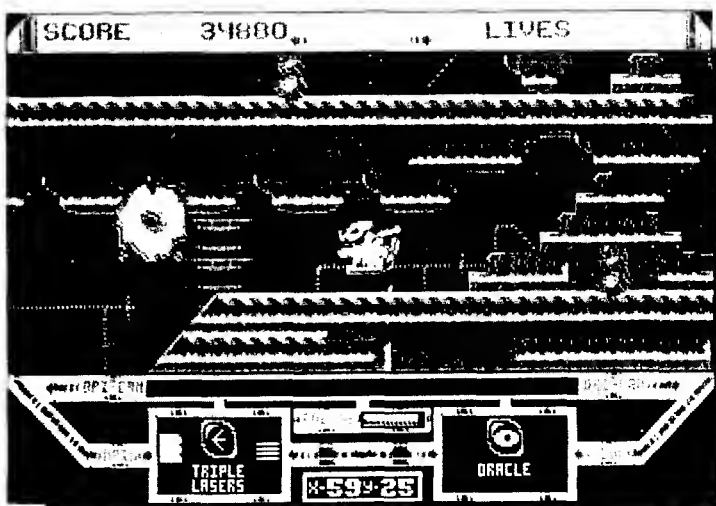
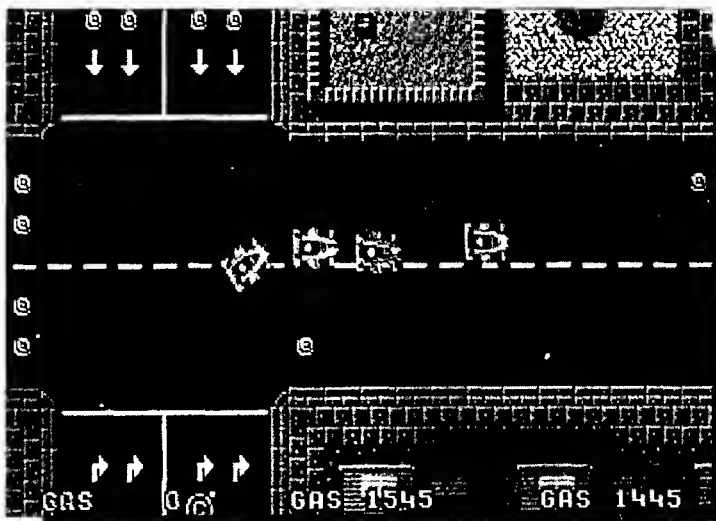
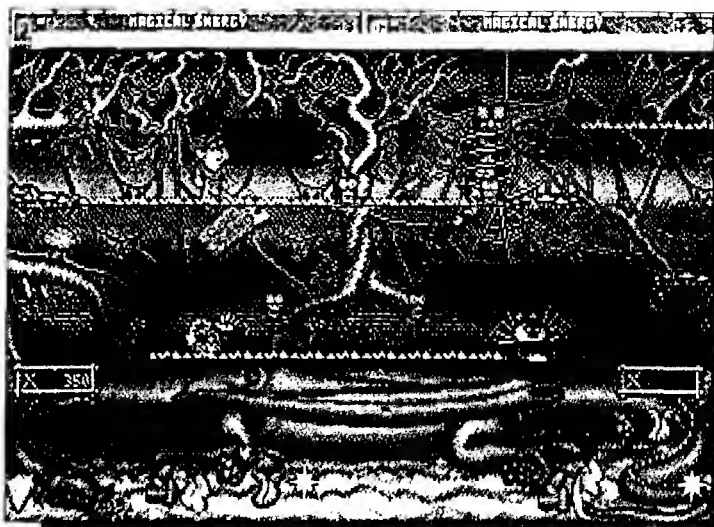
NITRO

A TURBO-hoz hasonlító programban felülről láthatunk egy gyilkos autóversenyt. A feladat életben maradni, amíg a verseny véget nem ér, és ha te futsz be elsőként az még jobb. Ez persze nem olyan egyszerű, mert az ellenfelek általában nem túl rendes emberek (nem is beszélve a computerről). A továbbjutáshoz minden módszert használhatsz, és az autók különböző módszerekkel "fejleszthetők"... Talán a legjobb, hogy extra pontokat kapsz, ha elülsz egy gyalogost. Állítólag egy ilyen matricát mellékelnek minden programhoz: "Túl sok gyalogos - túl kevés idő..."

Talán nem a cég legnagyobb erőfeszítései közé sorolható ez a program, de mindenesetre nem szórakoztató, egynek nem rossz!

THE KILLING
GAME SHOW

Egyfajta változata a Running Man témának. A program jól demonstrálja, hogy a gyártó az ilyesfajta akciós játékok nagymestere. Talán a legkönnyebben játszható Psygnosis program, de azért nem túl könnyű. Az apokaliptikus játékban a feladatod, hogy élve kimenekülj. SZUPER grafika, és természetesen gyilkos sztereo zene. A programba építettek egy ismétlés opciót is, amivel visszanezetheted az utolsó szinten elkövetett hibáidat.





Azt mondják, nem is érdemes azzal foglalkozni, hogy mi újság a nagyvilágban, ha a dolog itthon elérhetetlen. Lássuk mi jelent újat az itthoni swappereknek (akiknek egyébként minden hamarabb van meg, mint mi azt a nyugati újságokban olvassuk...).

Night Breed

Egy horrorfilm számítógépes változata. Mint ahogy az ilyen programoknál lenni szokott, nem valami jó, de eladható, mert a filmnek sikere van. Autónkat a térképen kell irányítanunk, ahol a leggyakoribb esemény, hogy úttorlaszokba ütközünk, fél percenként kifogy a benzinünk, stb. A legnagyobb kaland amiben aránylag hamar részed lehet, ha a térkép a kezdeti képernyőn lévő jobb felső részén található temetőbe becsattogsz. Itt egy vámpír úr kiszívja a véredet, ha nem futsz el. Elfutni a joystick teljes tönkretételével lehet, de nekünk nem sikerült... Biztos nem nagyon rossz játék, de ha nem láttad a filmet, akkor nem tudod például, hogy minek az a sok úttorlasz!



MURDERS IN SPACE

Egy űrbázison játszódik, szintén horrortörténet. A súlytalanság állapotában kell röpködnöd a bázison, és párbeszédet folytatni a figurákkal, akikkel találkozol. Ezekhez sok nyelv közül választhatsz a program elején, ha szerencséd van (vagy éppen nincs...) akkor valamelyiket beszéled. A grafika nem valami szép, a zene semmi, de egy HOT NEW játék!



STRIKE ACES

A Fighter Bomber nevű vadásszimulátor - akciójáték továbbfejlesztett változata. Tartalmaz egy trainer módot, és a küldetések választhatók lettek, de nem ad pontot. OK.

Kavalkád: Info A5000 !!!

Az angol újságokban megjelent egy cég (SSL), amely az Amiga 5000-et hirdeti. Mivel a hirdetésből csak az derült ki, hogy az A5000 gyorsabb, mint a 3000-es, és csak 400 GBP az ára, gyorsan kaptuk a telefont, amit egy nőszemély vett fel, aki azt sem tudta, hogy az Amigát eszik, vagy isszák. Természetesen küldtek nekünk egy információs csomagot, ami kellemes meglepetést okozott számunkra. A csomag alapján az SSL az első cég, aki az Amigák professzionális felhasználását lehetővé tevő bővítések gyártására, és profi marketingjére specializálódott. Mi nem írunk véleményt a három A5000-es modellről, hanem lefordítjuk a küldeményt, és közöljük, hiszen szerintünk sokakat érdekelhet.

Az A5000 információs csomag

Egy megvalósult amigás álom

Hosszú évekig kevesek privilégiuma volt a 32 bites számítástechnika. Most Te is csatlakozhatsz, és felfedezheted, hogy mi az amiért mások 2873 angol fontot fizetnek, (ti. A3000), az 500-asod, vagy 2000-esed álomgéppé alakításával, de olyan áron, amit Te is kibírsz!

Ár\teljesítmény áttörés

Elsőként az Amiga történelemben a Solid State Leisure áttörte az ár\teljesítmény gátat (ti. a 32 bites gépek hihetetlenül magas árát) egy teljes 32 bites belső bővítéssel (a processzor, a matematikai koprocesszor, a RAM, még az operációs rendszer is 32 bit-re vannak konfigurálva), amely még a 3000-est megveri teljesítményben.

Mire jó az A5000 ?

Ha egy komoly grafikai program felhasználó vagy:

Ha ray-tracing képet készítesz, ami 4 órát venne igénybe egy standard Amigán, 45 perc alatt kész lesz a 16.67 MHz-es A5000-rel matematikai koprocesszor nélkül, és a matematikai koprocesszorral (68881-16) 14 percet fog igénybe venni. Ez annyit jelent, hogy azt az animációt, amit eddig két hétig készítettél, most elkészítheted egy nap alatt.

Ha DTP felhasználó vagy:

Nem kell emlékeztessünk, milyen borzasztó lassúak a DTP programok egy standard Amigán (csak egy sor visszatörlése is nevetséges), de mindennek vége az új A5000-rel, hiszen 4MB superfast memóriát érhetsz el (a normális 9MB-os határon kívül) a hihetetlen 65 angol font per megabyte-os áron, és egy 5-600%-os sebességnövekedést a standard Amigához képest. Ez valóságos álommalá változtatja az Amiga DTP-t.

Ha Video Animátor vagy:

Nemcsak direkt módon gyorsul a munkád az A5000 használatával, hanem mert kb. 4 mp. animációt közvetlenül a superfast memóriából tudsz elérni. A Sculpt Animate 4D használata az A5000-rel biztosan mosolyt csal az arcodra...

Ha komoly szoftverfejlesztő vagy:

Szoftverfejlesztők számára az A5000 egy feltétlenül szükséges bővítés. A fordítási időt 5-600%-al gyorsítja fel, ami örömmel teszi a multi-tasking-ot. Így nem a Hard disk vár a C fordítódra, hanem fordítva! Ha pedig RAD-ot (Recoverable RAM disk) használasz, egy teljes fejlesztői környezetet tarthatsz a 32 bites superfast RAM-ban. Így ha a programod

Kavalkád: Info A5000 !!!

elGURUI, nem sokáig várnod, míg az Amigád felbootol, csak resetelsz és kész, máris dolgozhatsz tovább.

Ha tudományos munkát végzel:

Akkor tudod csak igazán teljesen kihasználni az A5000-et. A tudományos felhasználások területén minden programnyelvben használnak lebegőpontos könyvtárakat, amelyeket ha mellékelnek a hardver automatikusan használja a 68881\82IEEE 80-bites lebegőpontos matematikai koprocesszort, sőt egyes nyelveket, mint például az Absoft A\C Fortran\020-t direkt a 68020\68881-es kombinációjára tervezték.

Gondolj arra, hogy az A5000 teljesítménye megegyezik a SUN 3\100 munkaállomás teljesítményével. Az A5000 ugyanazt a CPU + FPU-t használja, mint a SUN, ugyanazokon a sebességeken (16-25 MHz), ugyanazt a 32 bites technológiát használja, beleértve a 32 bites nulla várakozási status-ú (zero wait states) RAM IC-eket.

Az A5000 vigan elfurikázik 2-3 MIPS-es sebességgel, és használ programokat, amelyeket a 68020\68881-es kombinációra optimalizáltak. Ez a sebesség 8 MIPS-ig növelhető "on chip cache" használatával. A 68882-50-es koprocesszor használatával elérheted a hihetetlen 1 MFLOPS sebességet, amivel egy standard Amiga 3 hónapos munkáját egyetlen nap alatt végezheted el.

Ingyenes SSL technikai\szoftver támogatás

Ha megvetted az A5000-det a cégünk INGYENES vásárlás utáni támogatásban részesít. Ez azt jelenti, hogy információt küldünk új termékek megjelenéséről, amelyek optimalizáltak az A5000 működéséhez, együtt egy 3.5"-es lemezzel. Az árban benne van a 12 hónapos garancia. Ez azt jelenti, hogy bármilyen rendellenességet észlelsz az A5000-rel, mi küldünk neked egy újat.

Legközelebb Londonban találkozhatunk a júniusi Ami-EXPO SHOW-n. Itt bemutatjuk a nagyteljesítményű grafikai munkaállomásainkat, a GRAFIX MACHINE-t, és az A3000-UX-et.

Az árlista (angol fontban):

A5000 @ 16.67 MHz + 1Mb RAM 399.00 + 2Mb RAM 468.00 + 3Mb RAM 534.00 + 4Mb RAM 594.00	A5000 @ 20.00 MHz + 1Mb RAM 499.00 + 2Mb RAM 568.00 + 3Mb RAM 634.00 + 4Mb RAM 694.00	A5000 @ 25.17 MHz + 1Mb RAM 599.00 + 2Mb RAM 668.00 + 3Mb RAM 734.00 + 4Mb RAM 794.00
WITH 68881 @ 16.67 MHz + 1Mb RAM 488.00 + 2Mb RAM 557.00 + 3Mb RAM 623.00 + 4Mb RAM 683.00	WITH 68881 @ 16.67 MHz + 1Mb RAM 588.00 + 2Mb RAM 657.00 + 3Mb RAM 723.00 + 4Mb RAM 783.00	WITH 68881 @ 16.67 MHz + 1Mb RAM 688.00 + 2Mb RAM 757.00 + 3Mb RAM 823.00 + 4Mb RAM 883.00
WITH 68881 @ 20.00 MHz + 1Mb RAM 509.00 + 2Mb RAM 578.00 + 3Mb RAM 644.00 + 4Mb RAM 704.00	WITH 68881 @ 20.00 MHz + 1Mb RAM 609.00 + 2Mb RAM 678.00 + 3Mb RAM 744.00 + 4Mb RAM 804.00	WITH 68881 @ 20.00 MHz + 1Mb RAM 709.00 + 2Mb RAM 778.00 + 3Mb RAM 844.00 + 4Mb RAM 904.00
WITH 68881 @ 25.17 MHz + 1Mb RAM 538.00 + 2Mb RAM 607.00 + 3Mb RAM 673.00 + 4Mb RAM 733.00	WITH 68881 @ 25.17 MHz + 1Mb RAM 638.00 + 2Mb RAM 707.00 + 3Mb RAM 773.00 + 4Mb RAM 833.00	WITH 68881 @ 25.17 MHz + 1Mb RAM 738.00 + 2Mb RAM 807.00 + 3Mb RAM 873.00 + 4Mb RAM 933.00
WITH 68882 @ 25.17 MHz + 1Mb RAM 604.00 + 2Mb RAM 673.00 + 3Mb RAM 739.00 + 4Mb RAM 799.00	WITH 68882 @ 25.17 MHz + 1Mb RAM 704.00 + 2Mb RAM 773.00 + 3Mb RAM 839.00 + 4Mb RAM 899.00	WITH 68882 @ 25.17 MHz + 1Mb RAM 804.00 + 2Mb RAM 873.00 + 3Mb RAM 939.00 + 4Mb RAM 999.00
WITH 68882 @ 33.00 MHz + 1Mb RAM 644.00 + 2Mb RAM 713.00 + 3Mb RAM 779.00 + 4Mb RAM 839.00	WITH 68882 @ 33.00 MHz + 1Mb RAM 744.00 + 2Mb RAM 813.00 + 3Mb RAM 879.00 + 4Mb RAM 939.00	WITH 68882 @ 33.00 MHz + 1Mb RAM 844.00 + 2Mb RAM 913.00 + 3Mb RAM 979.00 + 4Mb RAM 1039.00

Kavalkád: On Line Csomagrádió

Az első AM-ben olvashattak az érdeklődők a telefonos BBS-ek rejtelméről. Gondolom sokunk fantáziáját megmozgatta a cikk. A mostani írással még nagyobb mozgásra szeretném bírni a nyájas olvasó szürkeállományát. Egy olyan digitális - világméretű - adatátviteli hálózat lesz a téma, melyet rádióamatőrök hoztak létre hobby célból. Alapvetően nem kell sok egy ilyen rendszer kiépítéséhez. Hozzávetőlegesen: tízezer célszámítógép, pár ezer PC (kb. 100 gigabyte háttértárral), néhány műhold, na és persze jópár adóvevő. Ha ezek mind együtt vannak, már csak a lelkesedés és kitartás kérdése az egész. Úgy látszik a rádióamatőröknek mindez összességben.

Már régen vége annak az időnek, amikor a rádióamatőr a csöves rádiója mellett ülte ültögette a morse billentyűt és figyelte a sercegő rádiót. Mára rendkívül sok ága alakult ki a rádióamatőröknek. Csak néhányat ízeletül: nagytávolságú beszéd összeköttetés (a Föld bármely pontjával) állóképek átvitele (SSTV - Slow Scan Television - a szerk.), újabban mozgóképek átvitele amatőr berendezésekkel, rádióiránymérés ('rókavadászat'), a Holdra irányított rádióhullámokkal és az onnan visszaverődő jelekkel történő összeköttetés (Föld-Hold-Föld kísérlet), ugyanez a játék átvonuló meteorrajokra irányított rádióhullámokkal, hegytetőkön elhelyezett átvitelt (relé) állomások segítségével nagyobb távolságok áthidalása, műholdak fellövése, és azon reléállomások, mérőberendezések, kamera elhelyezése. És pár éve, számítógépek rádióon keresztül történő összeköttetése, azaz a CSOMAGRÁDIÓ.

1983-ban kezdődött az egész. Már előtte is létezett az úgynevezett RTTY (Radio Teletype), ami a telexezésnek felel meg, csak rádiók közbeiktatásával. 'Beszélgetésekre' megfelel, de az időnként előforduló vételi zavarok miatt adatvesztés léphet fel. A cél a hibamentes átvitel volt. Nem kellett feltalálni a spanyol viaszt. A gyakorlatban már működtek vezetékkel összekötött számítógépes hálózatok. Az X25 elnevezésű protokollal választották. Egy kis fejtoros után létrejött az AX25-re keresztelt protokoll, amely véletlen hozzáférésű csatornán, csomagkapcsolt adatátvitelt valósít meg.

A játékszabályok a következők:

- Egyszerre csak egy állomás adhat, az összes többi folyamatosan vételen van.
- Nincs kitüntetett állomás (server), mindenki azonos prioritású
- Az információt részekre bontva kell továbbítani
- Minden rész (csomag) ellenőrzőösszeggel van ellátva, így az adatátvitel közben keletkező hiba

kiszűrhető. Ha sérülés lép fel, nem kell teljes hosszban ismételni, hanem elegendő a hibás csomagot újraadni.

Az igazán izgalmas a dologban az a lehetőség, amit a számítógépek alkalmazása jelent. Ez teszi lehetővé, hogy valódi hálózatot lehessen létrehozni. Elegendő néhány kiemelkedő helyre telepíteni egy-egy adóvevőt és hozzákapcsolni egy célszámítógépet, ami az X25 protokollt ismeri. Már is létrejött a csomagkapcsolt rendszer.

A hálózatnak két fő eleme van. Az első a digitális átvitelt, ami az előbb említett célszámítógépből áll és általában hegytetőkre telepítik. A lényeg, hogy a szomszédos átvitelt vegyék egymás jeleit és tudjanak kommunikálni. Magyarországon 15 ilyen berendezést telepítettek a rádióamatőrök. Ezek leg többje az URH sávban működik (144.675 MHz) és vételi körzetük szinte az egész országot lefedi. Elegendő egy kisteljesítményű rádió, és máris hozzá lehet kapcsolódni a legközelebbi átvitelt. Azt utasítani tudom, hogy kapcsoljon tovább a következő csomópontba. Így több digitális átvitelt összeköttetésével megsokszorozható az áthidalható távolság. Ezt az összekapcsolhatóságot azonban nem a felhasználók szokták elsősorban kihasználni, hanem a hálózatot alkotó másik fontos építőelem a BBS-ek. Szerepük nagyon hasonló a telefonos társaikéhoz. Nagy háttértárral rendelkező számítógépek ezek, melyekben híreket, információkat, programokat, személyes üzeneteket lehet tárolni. A nagy különbség a telefonos és a rádióamatőr BBS-ek között, hogy az utóbbiak képesek egymást meghívni és automatikusan információt cserélni. A működtető szoftver biztosítja, hogy a környező BBS-ekkel rendszeres időközönként (óránként) megtörténjen a kapcsolatfelvétel. Ezek során a hírek (bulletin) és levelek (personal message) továbbításra kerülnek. Például ha egy angol rádióamatőrnek szeretné egy üzenetet küldeni, elegendő az egyik budapesti BBS-be betennem. (A rádióamatőr hívójelek egyértelműen azonosítják az országot. HG=Magyarország, G=Anglia, stb.) Amikor elérkezik az idő, a budapesti meghívja a bécsi BBS-t és közli vele, hogy egy levél van, amit Angliába kell továbbítani, majd elküldi az üzenetemet. A bécsi BBS is hasonló módon fogja tovább adni. Minden állomáson egy-két órát időzik az üzenet. Szerencsés esetben 1 nap alatt a címzett BBS-ében lehet a levél. Lehetőség van olyan címzésre is, amikor nincs konkrét végcél. Mindenkinek szól, közérdeklődő hírek különböző

Csomagrádió

BB KING

Kezdőknek

Cikk: BB King

Nagyon nehéz eltalálni azt a hangnemet, amit az ilyen típusú cikkekben meg kell ütni. Az AM első két száma alapján úgy gondoltuk nincs elég cikk az újságban, ami előmozdítaná a teljesen kezdő géptulajdonosok ismerkedését az új családdal az Amigával. Persze teljesen nem maradtak ki ezek a részek, de utólag végigolvasva azokat a cikkeket úgy ítéltük meg, hogy nem elég tiszták, érthetőek azok számára, akiknek még nem volt más számítógépük azelőtt, akiknek az is kínai, hogy CLI (Nekünk egyikünknek sem az első gépünk az AMI, "öreg fiúk" vagyunk...).

Megpróbáljuk egy kicsit tanfolyamszerűbbé tenni a dolgot, mindenkihez szólni. Mondhatod, hogy téged aztán nem érdekel, te játszani vetted a gépet, le... –od, hogy a gép mit csinál miközben te jól érzed magad. Hamarosan rá fogsz jönni, hogy egy csomó dolgot azért jó, ha te is tudsz. Az elsőként azt, hogy hogyan kell egy számítógépet (nem is feltétlenül Amigát) kezelni. Vannak olyan szabályok, amelyeket sokan nem tartanak be, és ezért nagy árat is fizetnek, például tönkremegy egy lemezük, de rossz esetben akár a gépük is. A legtöbb hiba persze javítható, de van olyan alkatrész az Amigában, ami bizony 1500 ATS-be kerül. Ha megtudsz egyet s mást a gépedről, már azt eléred, hogy nem tudnak minden baromságot rádszólni. Gondolj csak arra, hogy hány ember vesz belső memóriabővítőt sok pénzért. Erre sokaknak szükségük van, akik csak játszani akarnak is, hiszen egy csomó program csak 1MB memóriával fut. Ha egy kicsit tájékozódna előtte, kiderülne, hogy akinek 1.3-as gépe van (mostanában már csak ezt lehet kapni), annak a gépében van lehetőség memóriabővítésre mindenféle kártya nélkül is (az alaplapon). Ehhez csak a memória IC–ket kell megvenned, ami kb. 2800–3500 Ft. –ba kerül az 512K (4db). Ezt a négy memória IC–t csak be kell forrasztani a gépbe, és néhány műtyűrt átkötni (hogy a gép észre is vegye a változtatást). Ezt a műveletet sokan vállalják, akiknek a hírdetését megtalálod az újságban. Ezzel pénzt takarítasz meg, akkor is, ha egyébként CSAK játékra használnod azt a plusz memóriát.

Sok más példán keresztül bizonyítani tudnám, hogy mennyi előnye van annak, ha egy kicsit megismered a gépedet. Meglátjátok milyen jó érzés, ha valamit saját magad meg tudsz csinálni. Emlékszem az első számítógépprogramra, amit életemben írtam egy C64-en. Annyit csinált, hogy várt egy billentyűre, és ha bármit leütöttél kiírta, hogy "Gergő hülye, Bence pedig mérhetetlenül okos." (Gergő az öcsém). Akkoriban ő még semmit nem értett a gépekhez, úgyhogy könnyű volt neki bemesélni, hogy ez a gép saját véleménye, ő pedig már csak tudja!!! Egy ilyen programot bárki meg tud írni. Nem azt szeretném, hogy mindenkiből programozót faragjunk, (én például nem lettem jó programozó), de bizonyos feladatokat meg tudj oldani. Már minimális programozási ismeretek birtokában is másképpen nézed a profi programokat, ugyanúgy, mint ahogy másképpen hallgatja az ember Jimmy Hendrixet, ha már próbált elektromos gitáron játszani (arról nem is beszélve, hogy egy másik jó gitáros mennyivel másképpen hallgatja...).

Ezekben a cikkekben megpróbálom azoknak a fejével gondolkodni, akiknek a számítógép teljesen új, és gyakran

túl bonyolult, érthetetlen. Ha minden jól megy, remélem a gyorstalpaló után sok jó számítógép felhasználó gördül majd le az Amiga Magazin futószalagjáról, akik tudják majd, hogy mi micsoda, hogyan kell például egy vállalat könyvelését számítógépre venni, hogyan kell levelet nyomtatni, lemezeiket rendben tudják tartani, és még a programozásról sem hiszik majd azt, hogy az már varázslat! Ez a kis lelkesítő után nézzük mivel fogunk majd foglalkozni a jövőben:

- Általános tudnivalók, a számítógép működése
- A számítógép kezelése, hibamegoldások
- Az Amiga számítógép, speciális tulajdonságok, műszaki jellemzők
- Az Amigához kapható kiegészítők
- Az Amiga operációs rendszere (AmigaDOS, Workbench), egyszerű feladatok megoldása
- Bevezetés a számítógép-programozásba, a BASIC nyelvről

Ebben a sorozatban teljesen a NULLÁRÓL indulunk, legyen hozzá bátorságod. Ha először csak ezt a cikket is érted meg az egész újság "komoly" részéből, nem baj, később majd megérted a többi is, és hasznosnak találd azokat is.

Általános tudnivalók, a számítógép működése

A számítógép egy olyan masina, ami egy másodperc alatt sok-sok számítási műveletet el tud végezni. Két értéket össze tud hasonlítani, ezek alapján dönteni tud. Ezek a döntések mindig két lehetőség közül az egyiket fedik (pl. tovább folytassa-e a számolást vagy nem, stb.). Bonyolultabb döntések mindig egyszerűbbek sorozataként írhatóak fel. Ez a döntési lehetőség teszi képessé a számítógépeket arra, hogy programokat lehessen rájuk írni. Ilyen számítógép, és számítógép program működik minden kvarcórán, videomagnó programozható felvétel elindításában, mikrohullámú sütőben, stb. Egy kvarcórának a programja például a következőképpen működik:

Az óra lelke a kvarc. Ez egy másodperc alatt pontosan meghatározott számú jelet ad a CPU-nak (Central Processing Unit – központi egység). Ez attól függ, hogy mekkora a kvarc frekvenciája. Ha ez mondjuk 50, akkor a CPU minden 50–dik impulzus után felnöveli a másodpercek számát egyel. Itt jön be a számítógép döntési képessége. Megvizsgálja, hogy a másodpercek száma elérte-e a 60–at. HA nem, akkor újra várja míg eltelik 50 impulzus, és tovább növeli a kijelzett másodpercek számát, amíg el nem éri a 60–at. HA elérte a másodpercek száma a 60–at, akkor felemeli a percek számát egyel (megvizsgálja, hogy az elérte-e a 60–at), és nullázza a másodpercek számát. Erre a feladatra a számítógépben egy program szolgál. Ez egy nagyon egyszerű kis program, talán el is tudod képzelni, hogy, hogy néz ki, de azért majd a "bevezetés a programozásba" c. résznél majd erről bőven esik szó. A CPU a számítógép lelke, a központi egység. Ha már a kvarcóránál tartunk, a CPU utasítja például az LCD (folyadékkristályos) kijelzőt arra, hogy jelenítse meg a számokat, a CPU figyeli a nyomógombok helyzetét, stb. Hogy ezeket hogyan csinálja arról majd a következő számban... (Bocs BB–lap vége)

A Motorola 68040

Cikk: Compi

A Motorola 68040-es

Rögtön a cikk elején megkérem a hardware iránt érdeklődő, és érzékenyebb idegrendszerű olvasókat: lapozzanak. Ami most következik, az nem fehér embernek való. Ha mégis a cikk elolvasása mellett döntenek, akkor is érdemes a közelben elhelyezni egy vödört csurgó nyálunk számára, és állunkat fölkötni egy erős kendővel, mert ellenkező esetben menthetetlenül leesik. Hát igen ... A Motorola megint dobott egyet. Csak a változatosság kedvéért most kő helyett úthengerrel.

A nagy elődök

Kézzel való a 68000-es. Ez az a processzor, amit egy mezei amigás ismerhet. Ez volt a Motorola első 16 bites processzora. Fejlesztése 1977-ben fejeződött be, és bátran állíthatom, hogy 16 Mbyte maximális tárhajtatásával, 18 darab 32 bites belső regiszterével, és külön supervisor-user állapotával az akkori idők legjobb CPU-ja volt. Ha az IBM a PC sorozatában nem az Intel 8086-os család tagjait alkalmazza, ma talán nem is ismernénk az Intel nevet. A 68000-es család tagjai ugyanis kb. annnyival jobbak az Intel azonos kategóriájú processzorainál, mint amennyivel egy BMW felülmúlja a Trabantot. A 68000-es sikerén felbuzdulva a Motorola nekilátott a család továbbfejlesztésének. Így jött létre a szintén 16 bites 68010-es, majd a 32 bites 68020-as és 30-as.

Elméleti alapok

A Motorola 68000-es sorozata legfiatalabb tagjának boncolgatása előtt meg kell ismerkednünk néhány alapszakkal. Bizonyára sokan hallottak már a RISC processzorokról. A RISC processzorok lényege, hogy igen szűk utasításkészlettel rendelkeznek, de szinte minden utasítás végrehajtásához elegendő egy óraciklusnyi idő. Ezzel ellentétben az úgynevezett CISC processzorok (ilyen az A500-ban található 68000-es is) sok, összetett utasítást képesek végrehajtani, ez azonban együtt jár az egy utasítás végrehajtásához szükséges idő megnövekedésével. Pontosabban együtt járt a 68040-es megjelenéséig.

A mikroprocesszorok gyártási technológiája olyan rohamosan fejlődött az elmúlt években, amivel a memóriaáramkörök fejlesztése már nem tudott lépést tartani. Ez oda vezetett, hogy egy gyors mikroprocesszor ideje jelentős részét a memóriaáramkörökre várva töltötte el. Ennek kiküszöbölésére fejlesztették ki a chace (gyorsító) memóriákat. Ez egy 8-256 kbyte-nyi igen gyors, újabbban a processzorba integrált memóriaterület, amibe a processzor által használt memóriaterületeket előzetesen bemásolják. Ha a CPU-nak valamilyen adatra van szüksége a memóriából, előbb a chace memóriában keresi, és csak ha itt nincs benn, akkor fordul a külső memóriához. Természetesen nem tudható pontosan előre, hogy a processzor a következő pár utasítás során melyik memóriaterületről vesz majd be adatokat, így a chace vezérlési technikájától függ, hogy a CPU az adatok mekkora hányadát képes a chace-ből elővenni. Ezt nevezik találati aránynak.

Még egy fogalmat kell tisztáznunk, a pipeline szerkezet fogalmát. Hogy a mikroprocesszor elvégezhesse feladatát,

egy utasítás végrehajtásához számos műveletnek kell megtörténnie. Először is elő kell állítani az adott utasítás végrehajtásához szükséges vezérlőjeleket, azaz dekódolni kell az utasítást. Csak ez után jöhet az utasításban előírt műveletek elvégzése. Ha egy utasítás dekódolását már az előző utasítás végrehajtása alatt elvégezzük, időt takarítunk meg. Ez a pipeline (csővezetékrendszer) feladata. Míg a CPU egyik része egy utasítás végrehajtásán dolgozik, addig más részei már a következő utasításokat dekódolják. Az egész olyan mint egy valódi csővezeték: az elején befolyanak a feldolgozandó utasítások, és a végén ugyancsak már teljesen dekódolva folynak ki. Van azonban egy kis nehézség, éspedig az ugró utasítások. Ha a processzor egy feltételes ugró utasítást hajt végre, nem tudható 5-6 utasítással előbb, hogy mi lesz az ugrás után következő utasítás. Ezt a 68040-esnél egy igen elegáns módszerrel oldották meg. A pipeline-t kétágúra építették, és mihelyt kiderül, hogy egy feltételes ugró utasítás került a csővezetékbe, megkezdődik az ugrás után lehetséges mindkét ág behívása és előfeldolgozása.

Másik sebességnövelő tényező, hogy a CPU-ba nem egy, hanem két gyorsítótárat integráltak, egyiket a program, másikat az adatok számára. Ezzel olyan jó találati arányt értek el, hogy a 68040-esnek alig-alig kell a külső tárból behozandó adatra várakoznia.

Ennek a két trükknek, és a beépített lebegőpontos aritmetikai-logikai egységnek köszönhetően (ALU) elérték, hogy egy utasítást átlagosan 1.3 órajelciklus alatt hajtsanak végre a processzor. Csak összehasonlításképpen az Intel legújabb modelljénél az 486-osnál normál utasítások végrehajtásakor ez 1.5-szer gyorsabb, lebegőpontos számításoknál pedig 3.5-szer. Nyugodtan mondhatjuk tehát, hogy a 68040-es egy RISC processzor sebességét, és egy CISC processzor utasításkészletét egyesíti magában, mivel természetesen felülről kompatibilis az alap 68000-esssel.

Az processzor gyorsabb működését az is segíti, hogy az ALU-knak a processzor többi részéhez képest kétszeres frekvenciájú órajelet adtak. Ha mondjuk az alapórajel 25 Mhz, az aritmetikai és logikai egységek (ahol a tényleges matematikai műveletek lezajlanak) 50 Mhz frekvenciájú órajelet kapnak. Természetesen ahogy már a család legelső tagjánál, a 68000-esnél is, úgy itt is külön ALU foglalkozik az adatok és a címek kiszámításával, így miközben az adat ALU mondjuk két regiszter összeadását végzi, addig a cím ALU a következő utasítások valamelyikének operandus címét számolhatja.

Nos hogy hova vezet mindez? A 68040-es (még leírni is hosszú) kereken 20.000.000 utasítást képes feldolgozni másodpercenként, és ugyancsak egy másodperc alatt 3.500.000 lebegőpontos számítást végezhet.

A legújabb hírek szerint a Commodore cég a most fejlesztés alatt álló Amiga 4000-es gépe már a 68040-es mikroprocesszorral épül majd. Ha ez igaz, egy olyan gépet vehet majd bárki, ami egy Amiga 3000-es dobozában, egy kétszobányi szuperszámítógép teljesítményét nyújtja. De mi lesz, ha megjelenik a 68050-es?

Bódy Attila

HW – programozás

Rovatvezető: URZ & Compi

A DMA időzítések

A DMA – mint azt a kedves olvasó már bizonyára tudja – egy olyan módszer, amellyel a számítógép tárában a mikroprocesszor igénybevétele nélkül adatokat mozgathatunk. Az adatmozgatás történhet memória–memória (pld. blitter), memória–periféria (pld. képfrissítés), és periféria–memória (pld. diszk) között. Mivel az Amiga számítógépekben meglehetősen sok eszköz használja a DMA-t, ezért a rendszebusz időzítése igen zsúfolt. Az Amiga memória IC-i 120 ns alatt képesek egy írási vagy egy olvasási művelet elvégzésére. A gép 68000-es processzora 7.28 MHz frekvenciával ketyeg, és egy memóriahozzáférés legkevesebb négy órajelciklust vesz igénybe. Ez azt jelenti, hogy annyi idő alatt, amennyi alatt a processzor egy adatot képes fogadni a buszról, a memória két adatot is képes kibocsátani. Ezt használták ki a gép tervezői a DMA sebességének növelésére. A rendszer egy 68000-es memóriaciklus alatt két belső memóriaciklust hajt végre: az első a soronkövetkező DMA műveletet, a második a processzort szolgálja ki. Ez azonban így túl szép ahhoz, hogy igaz legyen. Ezt a képet a valóságban rendesen megkavarhatják a

Prioritások

Mi történik akkor, ha egy DMA eszköz olyan gyorsan igényli az adatokat a memóriából, hogy az egy ide, egy oda jellegű felosztásnál már nem kapná meg őket a számára szükséges sebességgel? Ha mondjuk ez az eset a képernyő DMA esetében fordulna elő, a meg nem kapott adat helyén szemét jelenne meg a képen, ami igencsak zavaróan hatna. Ezért hát ha egy egységnek olyan sok adata van szüksége, hogy a szokásos 1–1 ciklusos felosztással már nem szolgálható ki, akkor a rendszer a processzor számára nem ad ki adatot egészen addig, amíg a buszon nem lesz egy szabad memóriaciklusnyi hely. Ezért lassul le a gépünk például a 16 színű HIREs üzemmódban, mert ilyenkor a képernyő DMA olyan sok adatot igényel, hogy a processzor számára csak a sor, és képviesszafutás ideje alatt maradnak szabad memóriaciklusok. A DMA egységekről általában elmondható, hogy elsőbbséget élveznek a processzorral szemben. Ez alól egyedül a BLITTER kivétel, mert itt beállíthatjuk a prioritást. Erre szolgál a DMA CON regiszter BLTPRI bitje, amit ha egyre állítunk, a blitter elsőbbséget élvez a CPU felett, ha nullára, akkor pedig egyenlő lehetőségük van a busz használatára, azaz a blitter minden saját memóriahozzáférése után lehetőséget ad a processzornak egy memóriaműveletre.

A DMA időzítések pontos ismertetése előtt, nézzük mi használ DMA-t az Amigában. Az eszközöket prioritási sorrendben soroljuk fel. Ha egy magasabb és egy alacsonyabb prioritású eszköz egyszerre tart igényt a busz használatára, akkor a magasabb prioritású kapja meg.

Memóriafrissítés: Az Amigában dinamikus memóriák találhatók. Az ilyen fajta memória a belcirt adatot nem őrzi meg a végtelenségig, azt bizonyos időnként fel kell frissíteni. Hogy az adatok ne vesszenek el, minden rasztorsorban az első néhány memóriaciklust a frissítő logika kapja meg.

Disk DMA: A frissítés megtörténte után a lemezvezérlő használhatja a buszt három memóriaműveletnyi időre.

Audio DMA: Minden rasztorsorban minden hangcsatorna egy memóriaciklust kaphat a DA konverterre küldendő adat elővételére.

Bitplane DMA: A képernyő adatait a DDFSTRT regiszter által meghatározott pozíciótól kezdve hívja le a megjelenítéshez. LORES módban négy, HIREs módban kettőnél több bitplane esetén már az alacsonyabb prioritású eszközöktől vesz el egy-egy memóriaciklust.

Sprite DMA: A sprite-ok adatainak elővétele a sorviesszafutás megkezdése utáni huszonkettedik memóriaciklusban kezdődik, és legfeljebb az ötvenkettedik memóriaciklusig tart. Azért legfeljebb, mert ha a DDFSTRT regisztert \$38-nál kisebbre állítjuk, a képernyő DMA korábban kezd el adatokat lehívni, mint ahogy a sprite adatok lehívása befejeződött volna. Ez azt jelenti, hogy azok a sprite-ok, amelyeket nem tudott a DMA feltölteni, nem használhatóak fel. Ne lepődjön meg tehát senki sem azon, hogyha OVERSCAN képet használ, a felhasználható sprite-ok száma lecsökken. (Hogy mennyire, az attól függ, hogy mennyivel visszük \$38 alá a DDFSTRT regiszter értékét.)

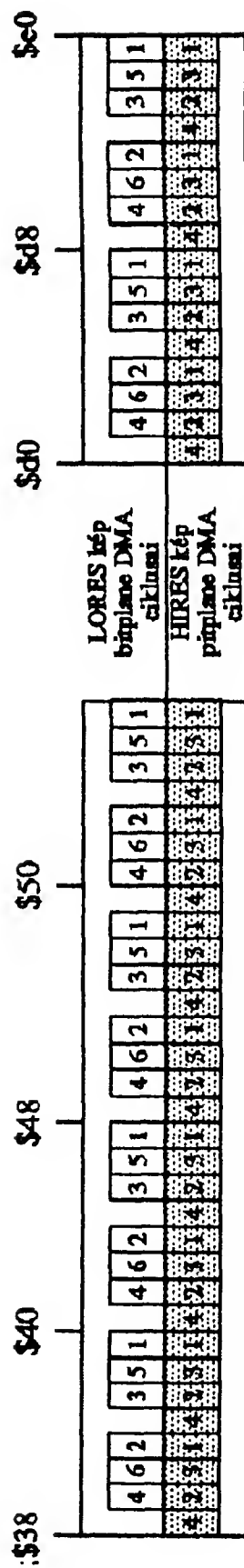
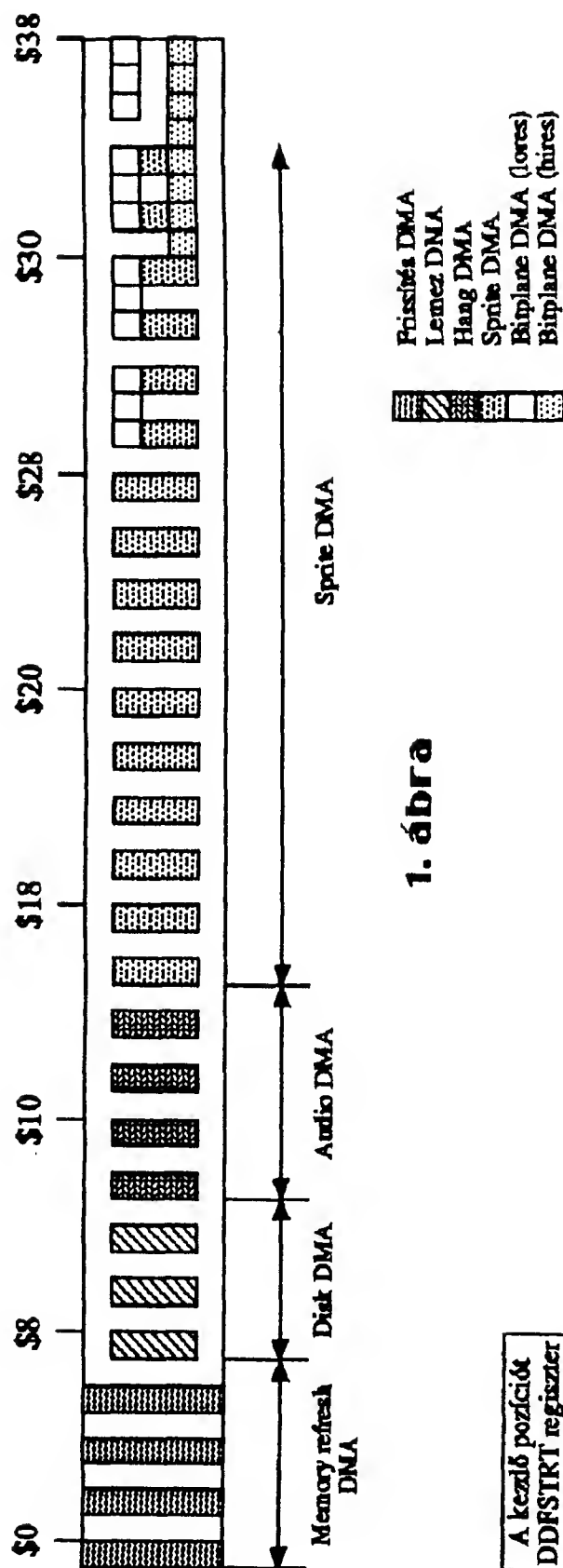
Copper DMA: A copper ezen a DMA csatornán veszi elő az utasításait a memóriából, és végzi el a kívánt műveleteket.

Blitter DMA: A blitter adatmozgatási műveleteinél használatos. A blitteren belül négy DMA csatorna található, ezek egy belső rendszer szerint osztoznak a buszon.

CPU: A 68000-es központi egység marad a legvégére. A processzort minden DMA egység várakoztathatja addig amíg szüksége van a buszra.

Az Amiga gépeken az egyes DMA eszközök a képernyőmegjelenítéssel összehangolva végzik munkájukat. Erre az átfedések elkerülése miatt van szükség. A sorviesszafutás idejére esnek azoknak az eszközöknek a DMA hozzáférései, amelyek a legfontosabb, nem halasztható feladatokat végzik el. Mivel a sorviesszafutás alatt a képernyővezérlő nem hív le adatokat a memóriából, így elkerülhető, hogy egy nagyobb prioritású eszköz esetleg akkor vegye igénybe a buszt, amikor a képernyővezérlőnek is szüksége lenne rá. Így a memóriافرissítés, a lemezegység és a hanggenerátorok DMA csatornái sohasem kerülnek konfliktusba a bitplane DMA-val. Más a helyzet a sprite-oknál. Ha a képernyő bal szélét a megszokottnál kijebb akarjuk húzni, a sprite DMA-nak már nem jut elegendő ideje arra, hogy az utolsó néhány sprite adatait elővegye, emiatt ezek a sprite-ok nem jeleníthetők meg.

Az Amiga DMA időzítése jól megfigyelhető az egyes és a kettes ábrán. Az előbbi a sorviesszafutás kezdetét követő, míg az utóbbi a képmegjelenítés alatti DMA hozzáféréseket mutatja. Az üres helyek szabadok a copper, a blitter, vagy a CPU memóriaműveleteihez. Az egyes DMA hozzáférések természetesen csak akkor hajódnak végre, ha az adott eszköz igényli őket. A 2. ábrán nyomon követhetjük azt is, hogy melyik bitplane melyik pillanatban veszi elő a számára szükséges adatot a memóriából, ha aktív.



Assembly kezdő

Rovatvezető: URZ

Előző számunkban a 68000-es regiszterkészletével ismerkedhettünk meg. Most –ígretünkhöz híven– a címzés módokat vesszük alaposabban szemügyre.

A 68000-esnek 14 féle címzés módja van. A címzés módok ismertetésekor nem térek ki a 68000-es család nagyobb tagjainál megengedett új címzés módokra, a 68010/20/30 plusz lehetőségeivel egy későbbi számban foglalkozunk majd. A címzés módokat nem érdemes 'bemagolni', de áttekintésük mindenképpen szükséges ahhoz, hogy felismerjük a logikai összefüggéseket, és később rutinszerűen használhassuk őket.

Regiszter közvetlen címzése

Ennél a címzés módnál az operandust vagy beírjuk, vagy kiolvassuk az adott regiszterből. Regiszterként az adat- és címregiszterek, valamint az ellenőrző regiszterek adhatók meg (SR, CCR).

Adatregiszter közvetlen címzése

Egy adott adatregisztert címezhetünk meg.

Szintaxis: Dx

Példa: MOVE.W huhu,D0
MOVE.B D0,(A0)

Működés: Az operandus az adatregiszterbe íródik, vagy onnan olvasódik ki.

Címregiszter közvetlen címzése

Egy adott címregisztert címezhetünk meg.

Szintaxis: Ax

Példa: LEA huhu,A0
MOVEA.L hehe,A0

Működés: Az operandus a címregiszterbe íródik, vagy onnan olvasódik ki.

Regiszter közvetett címzése

E címzés módnál egy címregisztert mutatóként használunk, tehát az operandus a címregiszter által meghatározott címen lévő adat lesz. Amennyiben az operandus nem byte hosszúságú, akkor a címregiszter az operandus legelső 8 bitjére mutat.

Szintaxis: (Ax)

Példa: MOVE.W D0,(A5)
MOVE.B (A0),hihi

Működés: Az operandus a címregiszterben tárolt címen helyezkedik el. Pl: ha A5=\$1234, akkor a \$1234 címen lévő adat az operandus.

Regiszter közvetett címzése utólagos inkrementálással

A címzés mód az előzőtől annyiban különbözik, hogy az utasítás végrehajtásakor a címregiszter értéke a megadott operandushosszal megnövekszik (inkrementálódik).

Szintaxis: (Ax)+

Példa: MOVE.W huhu,(A0)+
MOVE.B D0,(A0)+

Működés: Id. előző, de pl. szó hosszúságú adat esetén a címregiszter értéke kettővel nő.

Regiszter közvetett címzése előzetes dekrementálással

Az előzőhöz hasonló címzés mód, azzal a különbséggel, hogy a címregiszter értéke az utasítás végrehajtása előtt az operandushosszal csökken.

Szintaxis: -(Ax)

Példa: MOVEM.W A0-A5,-(A7)
MOVE.B D0,-(A0)

Működés: Id. közvetett címzés, de pl. longword hosszúságú adat esetén a címregiszter értéke négyvel csökken.

Regiszter közvetett címzése címkülönbséggel

Ennél a címzésmódnál a címregiszter és a 16 bites (előjeles) offset összege mutat a tulajdonképpeni operandusra.

Szintaxis: $\text{offset}_n(Ax)$

Példa: `MOVE.W huhu,16(A0)`
`MOVE.B D0,-26(A0)`

Működés: pl. ha az első példában $A0=1234$, akkor az operandus az $1234+16=1250$ címen helyezkedik el.

Regiszter közvetett címzése címkülönbséggel és indexszel

Ennél a címzésmódnál a címregiszter, a 8 bites (előjeles) offset és az indexregiszter összege határozza meg az operandust.

Szintaxis: $\text{offset}_n(Ax, R.X)$

Példa: `MOVE.W 3(A5,D7.W),D0`
`MOVE.B D0,-5(A0,D2)`

Működés: pl. ha az első példában $A5=1234$ és $D7=20$, akkor az operandus címe: $1234+20+3=1257$.

Programszámláló relatív címzése

Ennél a címzésmódnál a PC (Program Counter) és a 16 bites előjeles címkülönbség mutat az operandus legelső bytejára. Nulla az offset akkor, ha az operandus maga az offset.

Szintaxis: $\text{offset}_n(PC)$

Példa: `MOVE.W 14(PC),D0`
`MOVE.B D0,hihi(PC)`

Működés: az első példában az operandus $PC+14$ lesz.

Programszámláló közvetett címzése indexszel

E címzésmódnál a programszámlálóhoz hozzáadódik a 8 bites előjeles offset, valamint az indexregiszter tartalma a megadott hosszal.

Szintaxis: $\text{offset}_n(PC)(Rx.X)$

Példa: `MOVE.W 3(PC)(D0.L),D0`
`MOVE.B D0,-12(PC)(D7.W)`

Működés: ha az első példában $D0=1234$, akkor az operandus címe: $PC+1234+3$

Abszolút címzés szó hosszúságú címmel

Az operandus címét egy 16 bites előjeles szó adja meg, amely az operandus legelső bytejára mutat.

Szintaxis: $xxxx.W$

Példa: `MOVE.W 4.W,D0`
`MOVE.B $1234.W,(A0)`

Működés: második példánkban az 1234 címen lévő byte átmásolódik az A0 által mutatott címre is.

Abszolút címzés hosszúszó hosszúságú címmel

A címet egy hosszúszó adja meg, amelyet az operandus legelső bytejának címeként kell értelmezni.

Szintaxis: $xxxx.L$

Példa: `MOVE.W 4.L,D0`
`MOVE.B D0,$567890`

Működés: első példánkban a 4-es abszolút címen található szó bemásolódik D0 alsó 16 bitjére.

Adatok közvetlen címzése

Ennél a címzésmódnál az operandus az utasításban helyezkedik el, a processzornak nem kell regiszterhez vagy külső tárolóhoz fordulnia érte.

Szintaxis: $\#xxxx$

Példa: `MOVE.W #$12,D0`
`MOVE.B #123,12(A0)`

Működés: A # után megadott konstans értékek a második operandus által meghatározott helyre kerülnek.

Következő számunkban a tárkezeléssel, az adatmozgatással ismerkedünk meg. A MOVE (és bizonyos tekintetben rá hasonlító) utasítások alkalmazásával, tulajdonságaival foglalkozunk mélyrehatóbban.

URZ

C tanfolyam

Rovatvezető: Compi

Regiszterváltozók

A leggyakrabban használt függvények részletes tanulmányozása előtt meg kell még ismerkednünk két igen fontos változófajtaival. Az eddig ismertett változók közös tulajdonsága, hogy a memóriában helyezkednek el. Lehetőség van azonban arra, hogy néhány változót közvetlenül a processzor regiszteriben tartsunk. Miért jó ez nekünk? Tudnunk kell, hogy a memória elérése jóval lassabb, mint a regiszterké. Ezért lényeges sebességnövekedést érhetünk el, ha a leggyakrabban használt változókat regiszterváltozókként deklaráljuk. Ezt egyszerűen a deklarációnál a változó típusa elé írt register szóval tehetjük meg. Például:

```
register int i,j;
```

A regiszterváltozók használatakor két fontos szempontra kell tekintettel lennünk. Először is csak char, int, long típusú változókat, és ezek unsigned változatait, valamint mutatóváltozókat deklarálhatunk regiszterváltozókként. Ha más típusú változókat próbálunk meg így létrehozni, nem kapunk ugyan hibajelzést, de nem regiszter, hanem normál memóriaváltozó jön létre. Másodszor figyelembe kell vennünk, hogy az ilyen célra felhasználható regiszterek száma korlátozott, úgyhogy 5-6 regiszterváltozónál többet nemigen tudunk létrehozni.

Mutatók

A regiszterváltozókön kívül minden változó a memóriában tárolódik. Ebből következik, hogy minden változónak címe is van. A mutatóváltozókban egy-egy ilyen változó címét tárolhatjuk. A mutatóváltozók ugyanúgy típusal rendelkeznek mint a többi változófajta. Ez azt jelenti, hogy egy int típusú változó címét csak egy int típusú mutatóváltozóba tárolhatjuk el. Egy mutatóváltozót egyszerűen a változó neve elé tett csillaggal hozhatunk létre:

```
int *ip;
```

Ekkor létrejön egy ip nevű int típusú mutatóváltozó. Ha van egy k nevű változónk (ami szintén int típusú kell legyen), ennek címét a változó neve elé tett & jellel kaphatjuk meg. Tehát a k változó címét az ip mutatóváltozóba következőképpen helyezhetjük el:

```
ip=&k;
```

Ennek hatására a k változó memóriabeli címe az ip változóba kerül. Ha ezek után a k változó értékét át akarjuk másolni mondjuk egy n nevű (int) változóba, ezt kétféleképpen is megtehetjük. Legegyszerűbben természetesen az

```
n=k;
```

utasítással. De mivel a k címe már az ip mutatóban van, ezért k-t ip-n keresztül is elérhetjük:

```
n=*ip;
```

Ebből kitűnik, hogy ha egy mutatóváltozó egy (memória-) változóra mutat, a változóban lévő értéket a mutató neve elé tett csillaggal érhetjük el. Ezek szerint ip a mutatóváltozóban lévő címet, míg *ip a mutatóváltozóban lévő címen található értéket adja vissza. Nézzünk egy példát a mutatók használatára:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
char c,*cp;
```

```
c='*';
```

```
cp=&c;
```

```
putchar(c);
```

```
putchar(*cp);
```

```
}
```

Az eredmény:

**

A mutatók használata

Most a kedves olvasó bizonyára felteszi magának a kérdést, hogy vajon mire is jó ez az egész? A mutatók használata két területen a leggyakoribb. A C nyelvnek vannak olyan elemei, amelyek nem adhatók át függvényeknek, például a struktúrák. Mivel azonban a mutatók minden gond nélkül átadhatók, ezért az átadandó struktúra kezdőcímét elhelyezzük egy mutatóban, és ezt adjuk át a függvénynek. Gyakran előfordul az is, hogy olyan területekhez kell hozzáférnünk, ami nem a program saját területe (pld. képernyőmemória). Ekkor a terület kezdőcímét elhelyezzük egy mutatóba, és azon keresztül fordulunk hozzá.

Tömbök

A tömbök és a mutatók között igen sok hasonlóság tapasztalható. Pontosabban szólva összesen egy különbség van köztük. Egy tömb deklarálásakor két dolog történik: lefoglalásra kerül egy akkora memóriaterület, amekkora elég a tömb elemeinek elhelyezésére, majd létrejön egy mutatóváltozó is, amibe ennek a memóriaterületnek a címe kerül. Például a

```
char c[512];
```

deklaráció hatására a veremben lefoglalásra kerül egy 512 byte-os terület, aminek a kezdőcíme c mutatóváltozóba másolódik. Ha ezek után a c tömb 12. elemébe a 'h' karaktert akarjuk elhelyezni, ezt így tehetjük meg:

```
c[12]='h';
```

Mivel a c gyakorlatilag egy mutató, ezért ez az eljárás az ímént tárgyalt mutatóknál is felhasználható. Ha létrehozunk egy tömböt, és a címét átmásoljuk egy másik mutatóba, ezen a mutatón keresztül ugyan úgy elérhetjük a tömb elemeit, mint a tömb saját mutatóján keresztül:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
char c[128],*cp;
```

```
cp=c;
```

```
c[3]='b';
```

```
putchar(cp[3]);
```

```
}
```

Az eredmény:

b

Ezek szerint a tömb mutatóját ugyan úgy használhatjuk mint más mutatókat, tehát a c[0] kifejezés megegyezik a *c kifejezéssel, és &c[0] egyenlő c-vel. Két dolgot kell megjegyeznünk még a tömbökkel kapcsolatban: Az első, hogy a tömbök elemeinek számozása nullától indul, tehát az íménti c tömb a c[0] elemmel kezdődik. A másik, hogy amikor egy tömb valamelyik eleméhez fordulunk, nem ellenőrződik, hogy az index nem esik e túl a tömb létrehozásakor megadott határon. Ha például az előbbi c tömb 256. eleméhez akarunk hozzáférni, semmiféle hibajelzést nem kapunk. De mivel a tömb létrehozásakor csak 128 elem számára szükséges hely lett biztosítva, ezért jobbik esetben más változókat, a rosszabbikban pedig saját programunkat írhatjuk felül. Az ilyen hibák kiszűrése igen kellemetlen dolog, ezért az indexeléssel igen óvatosan bándjunk! A tömbök indexelésére nem csak konstans, hanem minden olyan kifejezés is felhasználható, ami egész (char, int, long) típusú eredményt ad:

```
main()
{
    float f[128];
    int i,j,k;

    i=12;
    j=2;
    k=24;

    f[i*j+k]=123.456;
}
```

String konstansok

Mi történik, ha a puts függvényt így hívjuk meg:
puts("Hello world!");

A puts függvénynek (ami egy szöveget ír ki a képernyőre) egy char típusú mutatót kell átadni, amiben a kiírandó szöveg kezdőcíme található, és a szöveg végét egy 0 karakter jelzi. A fordító, amikor egy függvény paraméterlistájában idézőjelek közé zárt szöveget talál, azt eltárolja a program egy bizonyos részén, utána biggyeszt egy nullát, és a függvénynek csak ennek a területnek a kezdőcímét adja át. Tehát ott, ahol char típusú mutató kell álljon, minden további nélkül állhat idézőjelbe tett szöveg is. Mivel a C nyelvben a stringeket nulla zárja le, ez a nulla a megadott szöveg végére automatikusan odakerül. Mint mindenre, erre is nézzünk egy példát!

```
#include <stdio.h>
main()
{
    char *c;

    c="Abakadabra";
    puts(c);
    puts("Csiribú-csiribá");
}
Az eredmény:
Abakadabra
Csiribú-csiribá
```

Fontosabb függvények

Szinte minden program fogad és küld adatokat a külvilág felé. A szabványos C nyelvben az adatok kijuttatására leggyakrabban a printf függvény használatos. Ezzel a függvénnyel a képernyőre kiíratunk szöveget, és változókat egyaránt. A printf függvénynek – más nyelvek kiírató utasításaitól eltérően – azt is meg kell adni, hogy a kiíratandó változó milyen típusú, és hogy milyen formában kell kiírni. Ezt a függvénynek átadott első paraméterben, az úgynevezett formátumvezérlő stringben kell meghatároznunk. A printf függvény hívásának általános alakja a következő:

```
printf(form_string,[[var1,var2,...,varn]]);
```

A form_string egy karaktermutató (char *form_string), és egy olyan karakterláncra mutat, ami kiíródik a képernyőre, miközben % jellel kezdődő formátumvezérlő jelek helyére a függvény paraméterlistájában megadott változók értékei kerülnek behelyettesítésre. Ez így talán egy kissé zavaros, de egy példa bemutatása után mindjárt érthetőbb lesz. Tegyük fel, hogy az i egy int típusú változó, és jelenleg a 1234-es értéket tartalmazza. A

```
printf("Az i változó értéke : %d.\n",i);
```

sor hatására a képernyőre az

Az i változó értéke : 1234.

szöveg fog kiíródni. Példánkban található egy jel ('\n') amiről eddig nem esett szó. A C nyelv szöveges konstansáiban a speciális jeleket egy földírt per jel után megadott karakterrel helyezhetjük el. Ilyen speciális jel többek között a soremelés ('\n') a tabulátor ('\t'). Tehát amikor a printf függvény a szövegben a '\n' (\$10) karakterrel találkozunk, a kiírást a következő sor elején folytatja.

Formátumvezérlés

Mivel a gyakorlatban nem csak integer típusú változókat kell kiíratnunk, ismerkedjünk meg a printf függvény formátumvezérlőivel. Ahogy az már az előzőekben kiderült, ezeket a formátumvezérlőket % jellel kell bevezetni. Ezek a egyrészt a kiírandó változó típusáról, másrészt a kiíratás módjáról tájékoztatják a printf függvényt. Az alábbi táblázat tartalmazza a legfontosabb formátumvezérlőket.

%c	char
%s	char[] (string)
%d	decimális int
%u	előjel nélküli int
%o	oktális int
%x	hexadecimális int
%f	fixpontos float*
%e	exponenciális float*
%ld	decimális long
%lu	előjel nélküli long
%lo	oktális long
%lx	hexadecimális long
%lf	fixpontos double*
%le	exponenciális double*

A csillaggal megjelölt eseteket az Aztec C fordító nem értelmezi. (???)

Ha numerikus változókat íratunk ki, lehetőségünk van a változó számára fenntartott karakterhelyek számának megadására is. A lebegőpontos (float, double) változóknál a tizedesjegyek száma is megadható. A

```
%6d
```

hatására a tízes számrendszerben kiírandó változó számára hat karakterhely lesz biztosítva a szövegben, és a ki nem használt helyek szóközzel töltődnek fel, míg a

```
%06d
```

esetében a ki nem használt helyekre nulla kerül. Lebegőpontos változók esetében ez kiegészíthető még egy paraméterrel, ami megadja, hogy az összes helyen belül, hány karakter álljon rendelkezésre a tizedesjegyek számára (pontosság). Ha egy float változót (a tizedespontot is beleértve) 10 karakterhelyen, két tizedesjegy pontossággal, bevezető nullák nélkül akarunk kiírni, a formátumot a következőképpen kell megadnunk:

```
%10.2f
```

Hogy minden teljesen világos legyen, lássunk egy kissé bonyolultabb példát a printf függvény alkalmazására!

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int i,j;
```

```
    i=1432;
```

```
    j=54;
```

```
    printf("i(dec)=%d i(ukt)=%o i(hex)=%x\n",i,i,i);
```

```
    printf("j(dec)=%d j(ukt)=%o j(hex)=%x\n",j,j,j);
```

```
}
```

Hatása:

```
i(dec)= 1432 i(ukt)= 2630 i(hex)= 598
```

```
j(dec)=000054 j(ukt)=000066 j(hex)=000036
```

Néhány tanács a printf használatához

A most következő tanácsokat saját érdekünkben tartsuk be, különben néhány meglepetés érhet bennünket! Pontosán annyi változót adjunk meg a printf függvény paraméterlistájában, ahány formátumvezérlőt elhelyezünk a szövegben! Ha több változót adunk meg nem íródik ki mind, ha kevesebbet a hiányzó helyett mindenféle zagyaság fog megjelenni. Ügyeljünk a változók típusára! Ha a változóra vonatkozó formátumvezérlő például %d, a változónak int-nek kell lennie, ellenkező esetben hibás érték nyomtatódik ki.

— folytatjuk —

base	edu	\$7f40e	
boots	dc 8	1205, 0	
boottrys	dc 1	\$5234015	
	dc 1	\$376	
prgstart	brs	realstart	
	dc 8	\$20, \$41, \$45, \$47	
	dc 8	\$54, \$45, \$45, \$52, \$20, \$20, \$49, \$52	
	dc 8	\$55, \$59, \$20, \$20, \$20, \$20, \$49, \$47	
	dc 8	\$24, \$45, \$45, \$54, \$45, \$52, \$4, \$56	
	dc 8	\$35, \$78, \$20, \$20, \$20, \$52, \$49, \$54	
	dc 8	\$45, \$45, \$24, \$20, \$4, \$47, \$47, \$47	
	dc 8	\$45, \$53, \$54, \$74, \$20, \$20, \$47, \$59	
	dc 8	\$54, \$45, \$57, \$47, \$52, \$52, \$49, \$47	
	dc 8	\$59, \$20, \$20, \$20, \$40, \$41, \$40, \$45	
	dc 8	\$52, \$20, \$45, \$78, \$74, \$65, \$72, \$60	
	dc 8	\$69, \$58, \$25, \$20, \$65, \$74, \$59, \$28	
realstart	MOVEA.L	4, A6	, execute base
	TST.L	\$226(A6)	, reset vektor ures ?
	REQ.S	initreset	, igen, inicializoles
	CMPI.L	#base+stbcifs, \$226(A6)	, reset vektor e mienk ?
	BEQ.S	initreset	, igen, inicializoles
	CLR.L	\$226(A6)	, new, reset vektor torlos
	MOVE.L	\$FC0004, \$20	, hideg reset
	TRAP	#0	
initreset	MOVEA.L	A4, A0	, bootstart kcm
	LEA	base, A1	
	MOVE.W	#100, D0	
copyprg	MOVE.L	(A1), (A1)+	
	DBF	D0, copyprg	
	MOVEA.L	4.W, A6	
	CLR.L	\$2E(A6)	
	MOVE.L	#base+stbcifs, \$226(A6)	, 7f40e
	JSR	-\$54(A6)	, sumukiddata
	MOVE.L	D0, \$234(A6)	, ellenorzo oszreg beall
	MOVE.L	#32C0C2C0, \$0FF:\$20	, color00,color01 = zold
	JSR	-\$50(A6)	, findreset
	MOVEA.L	D5, A1	
	MOVEA.L	\$:6(A0), \$0	
	MOVEQ	#0, D0	
	RTS		
dosname	DC.B	'dos library, 0	

Szerszámosláda

Rovatvezető: Compi

Ebben a rovatban olyan kisebb–nagyobb segédprogramokat teszünk közzé, amelyek egy–egy olyan problémát oldanak meg, amibe előbb utóbb mindenki beleütközik, aki nem csak játszik, hanem programot is ír az Amigán. A közölt programokat az általunk írtak, valamint az olvasóinktól a szerkesztőségbe beérkezettek közül választjuk ki.

Sorozatunk eddigi két részében megjelent két programmal azoknak kívántunk segítséget nyújtani, akik egy kész program autoboot-os verzióját szeretnék elkészíteni. Most egy olyan segédprogramot közlünk, ami a programfejlesztés jóval korábbi szakaszában könnyíti meg a programozó életét. Különbözik ez a program az előző kettőtől abban is, hogy míg az eddigiek kis méretük miatt elfértek egy számban, addig ezt két részben közöljük, mivel a rovat terjedelme kötött. E talán kissé hosszúra nyúlt bevezető után lássuk a medvét, azaz a programot!

A probléma

Bizonyára minden assembly-ban programot fejlesztő amigásnak gondot okozott már, hogy hogyan helyezze el a programban felhasznált grafikát. A legegyszerűbb, de ugyanakkor a legszerencsétlenebb megoldás, ha a grafikus adatokat a program egyenként tölti be a lemezről. Gondoljuk csak meg, ha a program csak tíz grafikai elemet használ, mindegyiket mondjuk nyolc fázissal, ez máris nyolcvan file beolvasását jelenti, ami hála az Amiga meglehetősen gyenge file–kezelő rendszerének, igen csak lassú folyamat lenne. Egy másik megoldás, hogy az összes grafika elemet összegyűjtjük egy file-ba, és egyben húzzuk be a memóriába. Ekkor természetesen tudnunk kell, hogy melyik elem hol kezdődik a file elejéhez képest. A legfrappánsabb megoldás, ha a grafikát fordítás közben fűzzük hozzá a programhoz. Ezt a módszert különösen vonzóvá teszi, hogy a DevPac assembler ismeri az INCBIN pseudo–utasítást, amivel egy file tartalma befördíthető a programba. Ha ezt az utat választjuk, feltétlenül gondoskodnunk kell arról, hogy programunk a chip–memóriába töltődjék, mert különben adatainkat a blitterrel nem érhetjük el. A grafikus adatokat ilyenkor is érdemes egy file-ba összefogni, hogy a fordítás gyorsabban menjen.

Még egy probléma

A másik gond, hogy milyen formátumban tároljuk el a grafikát. Kézenfekvő lenne az Amigánál megszokott módon, bitplane–ként tárolni. Ekkor előbb az első, majd a második, harmadik stb. bitplane teljes területét kell letárolnunk. Ennek a megoldásnak van azonban egy óriási hátránya. Ha az eltárolt alakzatot – a továbbiakban BOB (blitter object) – a blitterrel akarjuk kitenni a képre, a blittert annyiszor kell újra felprogramoznunk, ahány bitplane-t használunk. Hála azonban az Amiga rugalmas képernyőkezelésének, képünket felépíthetjük úgy is, hogy az egyes bitplane–ek azonos sorai egymás után helyezkedjenek el a memóriában. Ha a képernyőmemóriánk kezdőcíme ke , a bitplane–ek száma n , és egy bitplane egy sora b byte széles, akkor a modulókat $n*(b-1)-re$, az egyes bitplane–ek mutatóit pedig $ke+x*(n-1)-re$ kell állítanunk, ahol x az adott bitplane sorszáma (1–6). Ezzel a képernyőszervezéssel elérhetjük, hogy minden bob felrakásához a blittert csak egyszer kelljen felprogramoznunk. A két képernyőszervezés közti különbségek jól megfigyelhetők az 1. ábrán, ahol először a normál, majd az általunk ismertetett képernyő memóriabeli elhelyezkedését láthatjuk.

Hogy a gyorsabb képernyőkezelést biztosító formátumot használhassuk, a grafikus adatokat át kell konvertálnunk. Ezt teszi meg a most és a következő részben közreadandó program. A programnak a bemenet adatokat elő kell készítenünk. Először is a rajzprogrammal, amivel a grafikát kidolgoztuk, az egyes alakzatok fázisait egymás mellé, vagy alá kell raknunk úgy, hogy az egyes fázisok egy téglalapot töltsenek ki. Ha van mondjuk egy bob–unk nyolc fázissal, akkor mindet elhelyezhetjük egy sorba, vagy nyolcba

soronként egy fázissal, de legegyszerűbb ha két sorba helyezzük el őket soronként négy fázissal. Az így előkészített képrészletet IFP konverterrel vágjuk ki a képből, és mentjük ki bitplane–ként. Ezt procedúrát természetesen minden egyes alakzattal végig kell játszanunk. Ha ezzel megvagyunk, meg kell írni a parancsfile–t, amiből a konverterünk megtudja az egyes elemek nevét, méretét, hogy hány sorban és hány oszlopban helyeztük el őket, valamint hogy hány bitplane használatos az adott objektumnál. Ezt parancsorokkal kell megadnunk, minden sorban egy bob adatait elhelyezve. Az egyes parancssorok a következőképpen épülnek fel:

filename,xbobs,ybobs,bobsize,bobsize,planes

Az xbobs az egy sorban levő bobok, az ybobs pedig a bobsorok száma. A bobsize a bobok vízszintes mérete szavakban (pixel/16), míg a bobsize a bobok függőleges méretét tartalmazza. A planes paraméterrel adjuk meg, hogy az objektum hány bitplane–t használ. (4 szín–2, 8 szín–3, 16 szín–4 stb.) Ezeket a paramétereket nem kell mind megadni, ha valamelyiket kihagyjuk, automatikusan az előzővel dolgozik tovább a program. Példaként álljon itt a DEATHFIGHTER grafikai konvertálásánál használt parancsfile:

P Graphics converter V 2.3 by Compi 1991.*

```
nomask
status,1,1,20,48
digits,11,1,1,27
charset,20,3,1,22
eyes,9,1,2,32
guns,9
locks,9
flashes,9
bases,4
shields,2
mask
maps,8
enemys,6
ears,2
bobs,4
expls,8

#include <stdio.h>
#include <ctype.h>

#define nsize 20/*n words*/
#define ysize 48/*n lines*/

#define COM 1<<0
#define INF 1<<1
#define OUT 1<<2

LWORD *mob,*AllocMem();
FILE *inf,*out,*comf,*info,*fopen();
char mobnam[64],namebuf[64];
char *Comname="Graphics_Conv:convcom";
char *Inpath[128]="Graphics_Conv:inc";
char *Outname="ram:bobs";
char *Infname="ram:bobinfo.inc";
int normore=0,planes=4;
int actmaxx,actmaxy;
int mobysize=1,mobysize=10,xmob=1,ymob=1;
int error=0,maskfig=0;
```

Van két parancs ebben a file–ban, amiről eddig nem esett szó. A konverter képes ugyanis minden grafikai elemhez maszkot is készíteni. A maszkolást a mask parancssal be, a nomask parancssal pedig kikapcsolhatjuk. A maszkot a program közvetlenül a bob után helyezi el. Az egyes bob–ok helyére vonatkozó információkat a konvertálás után egy bobinfo.inc nevű file–ban találhatjuk meg, assembly nyelven. (pld. statusoffs equ 0) A konverter használata gyébként igen egyszerű. Indítása a

conv

vagy a

conv <commandfile> <inputpath> <outputfile> <infofile>
parancssal történhet a CLI–ből, ahol commandfile = a parancsfile neve, inputpath = a bemeneti adatok elérési útvonala, outputfile = a kimeneti file neve, infofile = az info file neve. Ha egyébként a programot egyszerűen csak conv–val indítjuk, kiíródik egy rövid használati utasítás.

Reméljük a conv sokat segít majd a játék, intro, stb. íróknak. Íme a program első része, természetesen C nyelven:

Assembly haladó

Rovatvezető: URZ & Compi

A most ismertetendő rutin egy viszonylag egyszerű, de mégis igen hatásos effektust valósít meg. Gyakran előfordul a programozói munka során, hogy egy címképet, vagy valamilyen más grafikát kell megjeleníteni a képernyőn. Ilyenkor gondot okozhat, hogy milyen módon "hívjuk elő" a képet. Ha egyszerűen csak kihajítjuk, az bizony elég mellbevágóan hat a program, vagy demó majdani szemlélőjére. Jóval kecsesebb, ha a színek fényességét fokozatosan növelve jelenítjük meg, mintegy előhozva ezzel a háttér szűrkeségéből. A kép levételével persze ugyanez a helyzet.

Hogy a különböző fényességű színek azonos sebességgel fel, illetve lehúzását elérjük, minden szín minden összetevőjére minden lépésben egyenes húzó algorit-mussal kell meghatározzuk az aktuális értéket. Az alább közölt programrészlet két hívható rutinja, a **colsinc** és a **colsdec** (az előbbi fel, az utóbbi pedig lehúzza a kép színeit) hívásakor az **a0**-ban várja egy struktúra kezdőcímét, amely a következő elemeket tartalmazza:

- 1 word : a szín kiindulási, ill. végértéke
- 1 word : belső felhasználásra
- 1 word : belső felhasználásra
- 1 longword : célcím (pl. egy színregiszter címe)

Ezt a struktúrát kell ismételnünk annyiszor, ahány színt akarunk változtatni. Az 1. szó \$ffff értéke jelzi, hogy a lista végére értünk.

Az itt közölt rutincsomag nem teljes, hiányzik belőle egy **lass** nevű rutin, ami nem tesz mást, csak vár egy bizonyos ideig. Célszerű ezt a várakozást a kép-frissítéssel szinkronba hozni, de egyszerűbb esetben megteszi egy dbf ciklus is.

Bódy Attila

* Coloreffect by Compi 1990 *

```

;----- a0=tablazat
colsdec    bsr      initdec
            moveq    #16-1,d0                ;counter
colsdec102 move.l    a0,-(sp)
            move.w    d0,-(sp)
            bsr      lass
            move.w    (sp)+,d0
colsdec100 movea.l    colptr(a0),a1
            move.w    curcol(a0),d2
            move.w    colcnt(a0),d3
            moveq     #3-1,d7
            moveq     #15,d6                ;color mask
            moveq     #1,d5                ;increment value
colsdec101 move.w    defcol(a0),d1
            and.w     d6,d1                ;defcol
            and.w     d6,d2                ;curcol
            and.w     d6,d3                ;counter
            move.w    d3,d4                ;save counter
            add.w     d1,d3                ;defcol+counter
            and.w     d6,d3
            cmp.w     d4,d3
            bhs      nodeccc
            sub.w     d5,d2
nodeccc     not.w     d6
            move.w    curcol(a0),d1
            and.w     d6,d1
            or.w      d1,d2
            move.w    d2,curcol(a0)
            move.w    colcnt(a0),d1
            and.w     d6,d1
            or.w      d1,d3
            move.w    d3,colcnt(a0)
            not.w     d6
            lsl.w     #4,d6
            lsl.w     #4,d5
            dbf       d7,colsdec101
            move.w    d2,(a1)
            adda.l     #nexttbl,a0
            tst.w     (a0)
            bpl       colsdec100
            move.l     (sp)+,a0
            dbf       d0,colsdec102
            rts

```

```
;----- a0=tablazat
```

```
colsinc    bsr      initinc
           moveq     #16-1,d0           ;counter
colsinc102 move.l    a0,-(sp)
           move.w     d0,-(sp)
           bsr      lss
           move.w     (sp)+,d0
colsinc100 movea.l   colptr(a0),a1
           move.w     curcol(a0),d2
           move.w     colcnt(a0),d3
           moveq     #3-1,d7
           moveq     #15,d6           ;color mask
           moveq     #1,d5           ;increment value
colsinc101 move.w     defcol(a0),d1
           and.w      d6,d1           ;defcol
           and.w      d6,d2           ;curcol
           and.w      d6,d3           ;counter
           move.w     d3,d4-         ;save counter
           add.w      d1,d3           ;defcol+counter
           and.w      d6,d3
           cmp.w      d4,d3
           bhs      noince
           add.w      d5,d2
noince     not.w      d6
           move.w     curcol(a0),d1
           and.w      d6,d1
           or.w       d1,d2
           move.w     d2,curcol(a0)
           move.w     colcnt(a0),d1
           and.w      d6,d1
           or.w       d1,d3
           move.w     d3,colcnt(a0)
           not.w      d6
           lsl.w      #4,d6
           lsl.w      #4,d5
           dbf        d7,colsinc101
           move.w     d2,(a1)
           adda.l     #nextctabl,a0
           tst.w      (a0)
           bpl       colsinc100
           move.l     (sp)+,a0
           dbf        d0,colsinc102
           rts
```

```
;----- a0=tablazat
```

```
initinc    movem.l   a0-a1,-(sp)
inincloop  clr.w      curcol(a0)
           clr.w      colcnt(a0)
           movea.l    colptr(a0),a1
           clr.w      (a1)
           adda.w     #nextctabl,a0
           tst.w      (a0)
           bpl.s      inincloop
           movem.l    (sp)+,a0-a1
           rts
```

```
;----- a0=tablazat
```

```
initdec    movem.l   a0-a1,-(sp)
indecloop  move.w     defcol(a0),curcol(a0)
           clr.w      colcnt(a0)
           movea.l    colptr(a0),a1
           move.w     defcol(a0),(a1)
           adda.w     #nextctabl,a0
           tst.w      (a0)
           bpl.s      indecloop
           movem.l    (sp)+,a0-a1
           rts
```

```
;-----
```

```
rsreset
defcol     rs.w       1
curcol     rs.w       1
colcnt     rs.w       1
colptr     rs.l       1
nextctabl  rs.w       0
```

```
; FOR EXAMPLE
```

```
testctabl  dc.w       $0f00          ;default value
           dc.w       $0             ;current value
           dc.w       $0             ;counter
           dc.l       dummy          ;pointer
           dc.w       $ffff          ;end of table
```



```
#include <stdio.h>
```

```
#include <exec/types.h>
```

```
#define xsize 20 /*in words*/
#define ysize 48 /*in lines*/
```

```
#define COM 1<<0
#define INF 1<<1
#define OUT 1<<2
```

```
UWORD *mob,*AllocMem();
FILE *inf,*outf,*conf,*infof,*fopen();
char mobnam[64],namepuff[64];
char *Comname=
    "Graphics_Conv:convcomm";
char Inpath[128]=
    "Graphics_Conv:inc/";
char *Outname="ram:mobs";
char *Infname="ram:mobinfo.inc";
int nomore=0,planes=4;
int actmaxx,actmaxy;
int mobxsize=1,mobysize=16;
int xmobs=1,ymobs=1;
int error=0,mskflg=0;
ULONG offset=0;
```

```
main(argc,argv)
int argc;
char **argv;
```

```
{
    char *namepuff,*label;
    char *np,*inpath,*iname;
    ULONG memsize;
```

```
if(argc>1) Comname=argv[1];
else {
    info("Usage: Conv [CommandFile] [InputPath]");
    info(" [OutputName] [Infname]\n");
    info("Defaults: CommandFile =");
    info(" Graphics_Conv:Convcomm\n");
    info(" InputPath =");
    info(" Graphics_Conv:Inc/\n");
}
```

```
if(argc>2) {
    inpath=argv[2];
    np=Inpath;
    while((*np++=*inpath++));
    np-=2;
    if(*np!=':' && *np!='/'){
        *++np='/';
        *++np='\0';
    }
}
```

```
if(argc>3) Outname=argv[3];
if(argc>4) Infname=argv[4];
if((conf=fopen(Comname,"r"))){
    if((infof=fopen(Infname,"w"))){
        if((outf=fopen(Outname,"w"))){
            do {
                getpars();
                if(!nomore){
                    memsize=mobxsize*mobysize*xmobs
                        *ymobs*planes*sizeof(UWORD);
                    if((mob=AllocMem(memsize,0L))){
```

```
namepuff=namepuff;
inpath=Inpath;
iname=mobnam;
while((*namepuff++=*inpath++)!='\0');
--namepuff;
if(*namepuff-1!=':' && *namepuff-1!='/'){
    *namepuff++='/';
    label=namepuff;
    while((*namepuff++=*iname++)!=0);
    info(mobnam);
    putchar('\n');
    if((inf=fopen(namepuff,"r"))){
        print(label,infof);
        print("offs\tequ\t",infof);
        writelong(offset,infof);
        putc('\n',infof);
        mobin(inf);
        mobout(outf);
        fclose(inf);
    } else {
        info("File ");
        info(namepuff);
        info(" not found.\n");
    }
    FreeMem(mob,memsize);
} else {
    info("There's not enough memory !\n");
    nomore=1;
}
}
```

```
while(!nomore);
print("mobsize\tequ\t",infof);
writelong(offset,infof);
putc('\n',infof);
fclose(outf);
} else info("Can't open output file.\n");
fclose(infof);
} else info("Can't open info file.\n");
fclose(conf);
} else info("Command file not found.\n");
}
```

```
/******
getword(fn)
FILE *fn;
{
    return((getc(fn)<<8)!getc(fn));
}
```

```
/******
info(string)
char *string;
{
    while(*string != '\0') putchar(*(string++));
}
```

```
/******
mobin(fn)
FILE *fn;
{
    int i,j;
    UWORD *mut;
    j=planes*mobxsize*mobysize*xmobs*ymobs;
    mut=mob;
```



Anubis Kft.
Budapest
Ferenciek Tere 4-8
1053
Tel.: 117-3877

Magyarországon a legolcsóbb !!!

Sőt, némely árucikkeket a bécsi ár alatt vásárolhat, illetve rendelhet meg nálunk !

- megtakarítja az utazás fáradalmait és költségeit
- forintért vásárolhat
- nagyobb értékű vásárlás esetén részletfizetési kedvezmény lehetséges
- viszonteladók számára egyedi megbeszélés szerinti hitelfeltételek
- törzsvevők részére kedvezményeket biztosítunk
- áraink a beszerzési árakat követik
- egyedi kéréseket is megpróbálunk teljesíteni
- magyarországi garanciaérvényesítés
- garancián kívüli gyors szervíz

Raktárról kapható:

NoName 3.5" DSDD lemez 750Ft (100 doboztól 712 Ft)

NoName 5.25" DSDD lemez 375Ft (1 doboz)

360Ft (10 doboz)

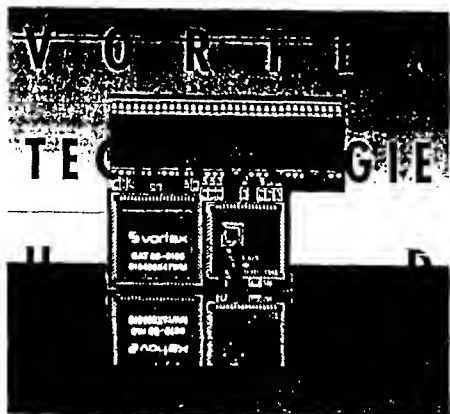
350Ft (50 doboz)

330Ft (100 doboz)

Ez Fantasztikus:

- 512K-s belső bővítő órával, kapcsolóval CSAK: 7580 Ft !!!
- 1.8 MB-os belső bővítő: 22.655 Ft

A500: 53.900 Ft !



ATonce AT emulator kártya - 42380 Ft

100% IBM AT kompatibilis Amiga, CGA, Herkules, Olivetti, Toshiba grafikus kártya, soros, és párhuzamos port emuláció. A gép képes lesz egyszerre futtatni Amiga és IBM programokat is.



ACTION REPLAY kártya - 11.500 Ft

Az AM januári számában ismertetett cartridge. Egyaránt hasznos a játékok könnyebb végigjátszásához, és hatalmas segítség a felhasználóknak is.



DIGIVIEW GOLD 4.0 - 24.750 Ft

A legjobb képdigitalizáló az Amigán. 768*580-as felbontásban digitalizál képeket 4096 színben kameráról. Képes 21 bites, és SuperHam képeket is digitalizálni.

MIDI interface - 5.635 Ft

Hangdigitalizáló - 9694 Ft

Megrendelhető:

AMIGA 2000:	144.650 Ft
Philips RGB monitor:	44.562 Ft
KCS POWER PC BOARD (XT kártya):	52.555 Ft
AMIGA 500-hoz 20 MB harddisk:	56.235 Ft
AMIGA 2000-hez 40MB harddisk:	93.955 Ft
PAL Genlock:	33.600 Ft

Visszacsalás

Ezt a lapot hajtsd meg a szaggatott vonal mentén, és küldd el amilyen gyorsan csak tudod !!!

Ezt a részt feltétlenül töltsd ki, ha visszaküldöd a lapot !!!

Név: _____ Város: _____ Lakcím: _____
Ir.szám: _____ Tel.: _____

Megrendelőlap

Mid van ?!

O-A500 O-A1000 O-A2000
O-A2500 O-A3000
O-egyéb: _____

O-512k O-1M O-1.5M O-2M
O-2M fölött

O-harddisk O-turbokártya
O-bridgeboard: _____

O-3.5" O-5.25" külső drive

O-genlock O-képdigitálizáló
O-egyéb: _____

O-Midi interfész O-hangdigitálizáló

Érdeklődési kör:

O-játéktesztek
O-játékíráások
O-software tesztek
O-software íráások
O-hardware íráások
O-Basic
O-C
O-68000 assembly
O-DTP
O-DTV

O-grafika, animáció
O-zene, MIDI
O-piaci információ
O-hírek, érdekességek
O-kapcsolat külföldi klubokkal
O-ON DISK
O-C64 csúcsprogramok
O-Atari ST melléklet
O-toplisták

(töltsd ki a név, cím rovatot !)

Megrendelem az AM-et _____ példányban
O-márciusi számot O-negyedévre O-félévre O-egy évre (148, 444, 888, 1776 Ft)

Apróhirdetés talon

Apróhirdetésed a következő számunkban ingyenesen közöljük ! (max. 30 szó)

Kérjük írd ide a véleményedet a lapról, ötleteid, esetleg kérdéseidet amik, ha közérdeklődésre tart számot, a következő számban megválaszoljuk. (ha tudjuk)

AM "Teccéyi" index

Kérjük karikázd be a megfelelő számot ! Véleményed maximálisan figyelembe vesszük !

1 2 3 4 5 Kavalkád
1 2 3 4 5 DTV
1 2 3 4 5 C tanfolyam
1 2 3 4 5 Szerszámosláda
1 2 3 4 5 Hardware-progr.

1 2 3 4 5 Assembly haladó
1 2 3 4 5 BASIC kezdő
1 2 3 4 5 AmigaDOS
1 2 3 4 5 Ray-Tracing
1 2 3 4 5

1 2 3 4 5 Mi újság ?
1 2 3 4 5 ON LINE
1 2 3 4 5 Virologia
1 2 3 4 5 Hardware
1 2 3 4 5 DTP

1 2 3 4 5 Zene
1 2 3 4 5 Börze
1 2 3 4 5 Plac
1 2 3 4 5 PD, ON DISK
1 2 3 4 5

1 2 3 4 5 AM válaszol
1 2 3 4 5 Játék
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

1 2 3 4 5 F18 leírás
1 2 3 4 5
1 2 3 4 5 6804u
1 2 3 4 5 A5000
1 2 3 4 5

1 2 3 4 5 AM kezdőknek AK
1 2 3 4 5 AM haladóknak
1 2 3 4 5 AM profiknak

A Te szinted: AK Hal. Prof.

5- csak így tovább 4- nem az igazi 3- elmegy 2- tragédia... 1- szüntessék meg !

Info

Bővebb információt kérek az AM-ben megjelent hírdetésről. A hirdetés száma bekarikázandó !

1 2 3 4 5 6 7 8 9 10

A név - cím rovat a másik oldalon !!!

ON DISK

Megrendelem az AM ON DISK lemezét.
(Név, cím a túloldalon !!!)

O- AM ON DISK Nr.1 - 200 Ft, utánvétellel.
(decemberi számhoz)

O- AM ON DISK Nr.2 - 200 Ft, utánvétellel
(januári számhoz)

----- 0 -----

O- negyed évre (500 Ft) - engedmény

O- fél évre (1000 Ft) - egy lemez ingyen

O- egy évre (1800 Ft) - három lemez ingyen

Az összeget rózsaszín postautalványon kell a részünkre feladni. Írják rá: ON DISK

Toplista szavazólap

A legjobb játék:

A legjobb felhasználói program:

Üres HELY

PD

Megrendelem az AM Public Domain lemezét.
(Név, cím a túloldalon !!!)

Szám	Tartalom	*
AM001	QRT (Quick Ray-Tracer)	
AM002	JRComm, MSH, NIB	

Utánvétellel * (3.5"): 200 Ft

Programozás: Modula 2

Az Amiga Magazin előző számában már jelent meg egy cikk a Modula-2 programnyelvről, de a cikk rövidsége miatt csak rövid bemutatásra kellet szorítkoznom. A most induló cikksorozatban részletesen próbálom majd írni a nyelvről, és használatáról az Amigán. Amennyire ezt meg lehet állapítani, Magyarországon az Avant Garde Software cég által készített Benchmark Modula-2 van leginkább "elterjedve" (ez az "elterjedés" feltehetőleg néhány tucat embert jelent csak). Valószínűleg azért is ilyen kevesen tartanak fenn néhány lemezt a programnak, mert itthon nincs a Modula-2 -höz (és pláne Amiga Modula-2 -höz) dokumentáció. Ezen próbálunk mi most segíteni, és remélem nem sikertelenül.

Ha nincs a birtokodban egy Modula-2 compiler, úgy érdemes egyet keríteni (nem lehetetlen), mert megéri. Ennek indoklását az előző számban már olvashattad, most itt csak annyit, hogy ami az IBM-nek Turbo Pascal, az az Amigának Modula-2. Ha valamilyen oktatási intézményben mindenáron arra kényszerítenek, hogy Turbo Pascal -t használj (igen, manapság is találkozhatunk ilyen erőszakos brutalitásokkal), akkor az ott tanultakat jól használhatod az Amigán is, ha Modula-2 -ben programozol. Itt jegyzem meg, hogy a sorozatban a Pascal nyelv alapszintű ismeretét feltételezem, de ha valakinek ez nincs me úgy egy Pascal tankönyvből könnyen megszerezheti (ezek itthon mindenhol beszerezhetők). A Modula-2 másik előnye, hogy kezöknek is elég egyszerű, ugyanakkor a profik igényeit is (általában) kielégíti (a Cinemaware cég Lords of the Rising Sun című játéka is Benchmark Modula-2 -ben készült). Nem folytatom most ezt az eszmefuttatást tovább, inkább hozzálatok a programnyelv leírásához, végtére is ez cikk feladata.

A Modula-2 programok modulokból épülnek fel. Mint már az előző számban leírtam, három féle modul létezik: meghatározó (definition module), megvalósító (implementation module), és program modul (program module, vagy röviden: module). Az első két modulfajtaival most nem foglalkozom, ezek ismerete és használatát kezdők számára nem szükséges. A program modul szintaktikai felépítése:

```
MODULE ModulNév;
```

```
...
```

```
END ModulNév.
```

Fontos most megjegyezni, hogy a Modula-2 különbséget tesz kis és nagybetű között. A nyelvben minden beépített utasítás nagybetűkből áll, ha az END helyett End -et írsz, úgy azt a fordító nem fogja elfogadni, és hibáüzenettel

leáll. A modulon belül az alábbi dolgok lehetnek (de nem feltétlenül szükségesek): import lista, konstans deklaráció, típus deklaráció, változó deklaráció, procedúrák és funkciók, és utasítások. Nézzük meg mindegyiket közelebbről!

Az import lista új azok számára, akik eddig Pascalban programoztak. Segítségével megadhatjuk a más modulokból felhasznált "importált" konstansokat, adattípusokat, változókat, procedúrákat. Formája a következő: FROM ModulNév IMPORT mit1, mit2, mit3, mit4, mit5;

,vagy ha az adott modulból mindent fel akarunk használni, úgy:

```
IMPORT ModulNév;
```

A két mód között az a különbség, hogy az első esetben a mit2 procedúrát így kell hívni:

```
mit2(a,b,c);
```

addig a második esetben ezt így kell megtenni:

```
ModulNév.mit2(a,b,c);
```

A Benchmark Modula-2 lemezen több tucat modulban adattípusok és procedúrák százai (összesen kb. 2500) található. Ezeket a modulokat a cég készítette, de a későbbiekben magunk is fogunk ilyeneket gyártani a saját céljainkra. A programnyelv moduláris felépítése miatt a "gyári" és a saját modulok között nincs különbség, saját modulkönyvtárunkat úgy használhatjuk, mintha az eredeti programcsomag része lenne.

A konstans deklarációban határozzuk meg a programban használt konstansokat és ezek értékeit, a Pascallal megegyező módon:

```
CONST
```

```
pi = 3.141592654;
```

```
e = 2.718281828;
```

```
száz = 100;
```

```
igen = "I";
```

```
...
```

Az itt meghatározott konstansok értéke a programban nem változtatható (azért konstans, és nem változó).

A típus deklaráció segítségével definiálhatunk új, virtuális adattípusokat. Formája a Pascallal megegyező:

```
TYPE
```

```
Tábla = ARRAY[1..8],[1..8] OF INTEGER;
```

```
egy tömb,
```

```
Bizonyítvány = RECORD
```

```
Matematika : INTEGER;
```

```
Fizika : INTEGER;
```

```
Magyar : INTEGER;
```

```
Átlag : REAL;
```

```
END;
```

```
egy rekord,
```

```
Színek = SET OF (Piros, Sárga, Kék, Zöld);
```

```
egy halmaz, és
```

Programozás: Modula 2

TáblaPtr = POINTER TO Tábla;
mutató esetében.

... Ezen kívül Modula-2 -ben lehetséges ún. "fejlett" adattípusokat (angolul Advanced Data Type, vagy röviden ADT) használni. Ezek csak a DEFINITION modulokban találhatók meg ilyen formájukban:

TYPE

Valami;

Itt most nem határoztuk meg, hogy a valami milyen típusú, ezt a munkát a megvalósító részre, az IMPLEMENTATION MODULE -ra hagytuk. Az ADT használata kezdők számára nem fontos, majd csak a későbbiekben fogunk rá visszatérni.

A változók deklarációja ismét csak a Pascal szintaktikával fog megegyezni:

VAR

a,b,c : REAL;

SakkTábla : Tábla;

Tűzijáték : Színek;

... A procedúrák ismét csak a Pascal rendszerűek, de a funkcióknál már eltérés van. Nézzük meg először egy procedúrát:

PROCEDURE Negatív(VAR x: REAL);

BEGIN

x := -x;

END Negatív;

Funkciók használatánál az eredmény visszajuttatásának módjában van eltérés:

PROCEDURE Negatív(x : REAL) : REAL;

BEGIN

RETURN (-x);

END Negatív;

Természetesen Modula-2 -ben is, mint

egy

Pascalban, a procedúráknak és a funkcióknak külön konstans, adattípus, és változó deklarációja lehet, de akár újabb procedúrákat és funkciókat is tartalmazhatnak magukba ágyazva:

PROCEDURE BlaBla;

CONST a = 1;

VAR b : INTEGER;

PROCEDURE Bla;

VAR R : REAL;

BEGIN

xxx

END Bla;

BEGIN

xxx

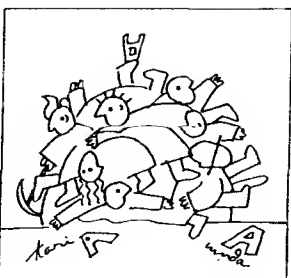
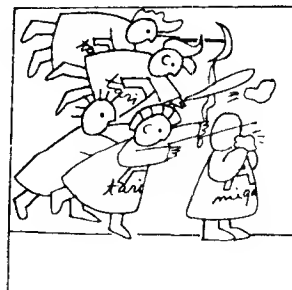
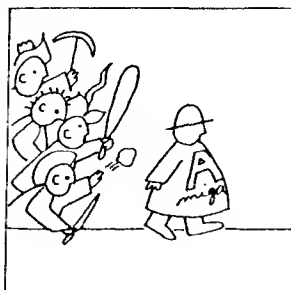
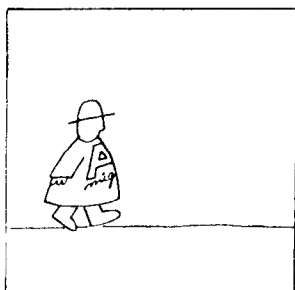
END BlaBla;

Ebben az esetben a Bla procedúra van a BlaBla procedúrába ágyazva.

Az utasítások a BEGIN kulcsszó után találhatóak, és az END ModulNév. utasításig tartanak (a "." fejezi be a programot). Az utasítások az alábbi típusúak lehetnek: a := b; , procedúrahívás, IF, CASE, WHILE, REPEAT, FOR, WITH, EXIT, és RETURN.

Ezzel végére is értünk a programnyelvnek, mert ez a teljes utasításkészlete. Mint látható, a Modula-2 egy nagyon kicsi programnyelv, valóban csak néhány utasítást tartalmaz, ezért is könnyű megtanulni. A programnyelv moduláris felépítése miatt azonban rendelkezésünkre áll további utasítások ezrei, melyek segítségével az Amiga teljes hardvere programozható. Az utasítások szintaktikája még hátravan, erről a következő hónapban részletesen írok majd. Addig is lehet utánajárni a Modula-2 compilemek, és átnézni az eddig tanultakat. Üdv,

Jazz



Programozás: MAD Trükkök

BOOTTRACK-es betöltő

Több barátom megpróbált meggátolni abban, hogy ezt a rutint leközöljem az AM-ben, mondván, hogy ezzel bárki olyat tud csinálni, mintha egy "rendes" program lenne. Én azonban már nagyon únom az olyan demo-kat, amiket egy CLI orientált menüből kell indítanod, és szana-szét GURU-zzák a buta arcukat.

A célunk egy olyan lemez készítése, amit ha betesz a drive-ba, illetve ha CTRL-AMIGA-AMIGA, akkor rögtön a megadott helyre tölti a programodat. Erre azért van szükség, hogy a lemezen ne kelljen a rendszernek semmilyen elemét rajta tartanod, ami fölöslegesen foglalja a helyet.

Megírod a programot, majd azt fix címre fordítod. A SEKA-ban a wt (write track) utasítással kiírod a lemez megadott TRACK-jeire a futtatni kívánt programodat.

Például:

```
wt
$30000
20
10
```

Ez az utasítás a megfelelő paraméterekkel a \$30000 címtől kezdődő memóriaterületet kimentti a 20-as TRACK-től kezdve, 10 TRACK-hosszan. Tehát előre ki kell számolnod, hogy a programod hány TRACK hosszú. Egy TRACK-re \$1600 byte információ fér (DOS formátumban...).

Ha ezzel kész vagy, akkor jön be a képbe a most közölt program. A rutin elején a futtatandó programod paramétereit kell beállítanod. Ezután fordítsd le, majd ki kell számoltatnod hozzá a CHECKSUM-ot. Ezt a MasterSEKA-ban a b paranccsal végezheted el.

Ezt írd be a fordítás után:

bCHECK

ezután mentsd ki a programot a lemez 0-dik TRACK, nulladik, és első SECTOR-ára.

```
wt
CHECK
0
1
```

Ezzel készen is vagy. Ha mindent jól csináltál, akkor egy olyan lemezt kapsz, amit ha BOOT-olsz, akkor a megadott programod azonnal betöltődik. További hozzáfűzni valóm:

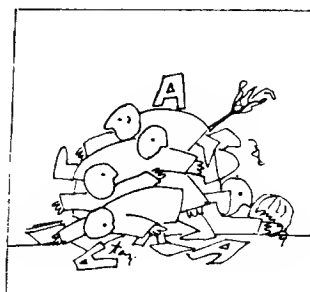
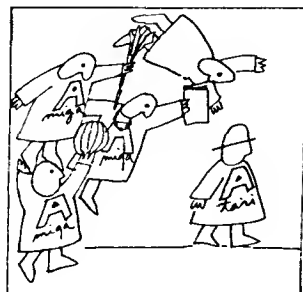
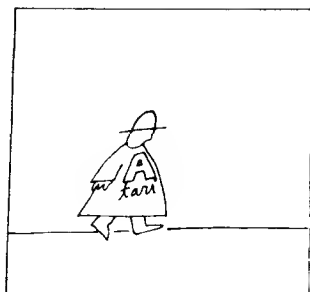
Ha töltés közben kiveszed a lemezt, akkor a képernyő pirosra változik. Természetesen ilyenkor a megfelelő TRACK rosszul töltődik be. Ha a lemezt visszateszed, a hibás TRACK újra betöltődik. Így ha a lemezt ki-be rángatod is, a programod akkor is helyesen töltődik be.

Az itt közölt verzió természetesen csak az alapvető dolgokat tartalmazza. Ezt a verziót bárki továbbfejlesztheti, például hozzáfűzhetsz egy billentyűzetlefigyelő menüt, vagy akármit. Ha az elindított Demoból reset-elsz, azonnal a menübe kerülsz vissza. Így színvonalasabb demo lemezeket készíthetsz.

Ezt a módszert használják azok a programok, amelyek csodák csodájára azonnal kitesznek egy menüt, vagy scroll-t, grafikát, stb, amint betesz a lemezt.

Végetért a móka mára, zárul MAD mókatára, de ha tetszett: akkor legközelebb is írok sok érdekes adatot, például, hogy a bálna az nem is hal, hanem madár, és banánnal táplálkozik.

THE MAD



Programozás: MAD Trükkök

```

kezdes:=$40000;HOL INDUL
trac:=$79000;DRIVE PUFER
track:=18;HANYADIK
CYLINDERTOL -2
start:=$40000;HOVA
TOLTSON
tolt: =20;HANYTRACKOTTO
LT
dc.w $444f,$5300,0,0,0,0;
k: lealo(pc),a0
   move.la0,$80
   trap#0;MEGSZAKITAS
GENERAL
; rts
jmpkezdes
lo: move.w#$8190,$dff096
; move.w#$4000,$dff09a;
MEGSZAKITAS LETILT
bsrkpo
ok: bsrdisk
    bsrco
    bsrlep
    add.l#$1600,d6
    addq.l#1,d0
    dbfd5,ok
    rte
lep: ;FEJ LEPTETES
     bclr#1,$bfd100
     eor.b#4,$bfd100
     btst#2,$bfd100
     beq.ski0
     bclr#0,$bfd100
     bset#0,$bfd100
kio: rts
kpo: move.b#track,d0
     move.l#tolt,d5
     move.l#start,d6
     move.b#$ff,$bfd100
     bsrvar
     bclr#7,$bfd100
     bsrvar
     bclr#3,$bfd100
po: btst#5,$bfe001
    bne.spo
    bset#1,$bfd100
    btst#4,$bfe001
    beq.sel
    bclr#0,$bfd100
    move.w#$2200,d7
kuk: nop
     nop
     dbfd7,kuk
el: bset#1,$bfd100
gizguz:bclr#0,$bfd100
     move.w#$0180,d7
go: nop

dbfd7,go
bset#0,$bfd100
btst#4,$bfe001
bne.sgzguz
bset#2,$bfd100
cmp.b#$00,d0
beq.skuku
cmp.b#$01,d0
beq.skas
clr.ld2
move.bd0,d2
lsr.w#1,d2
bcc.sat
bclr#2,$bfd100
at: bclr#1,$bfd100
kik: bclr#0,$bfd100
     bset#0,$bfd100
     move.w#$0780,d7
ret: nop
     dbfd7,ret
     dbfd2,kik
kuku:rts
kas: bclr#2,$bfd100
     bclr#1,$bfd100
     bclr#0,$bfd100
     bset#0,$bfd100
     move.w#$0700,d7
rut: nop
     dbfd7,rut
     rts
disk:bclr#1,$bfe001;PUFFER
TOLTES
     move.w#$8500,$dff09e
     move.w#$4489,$dff07e
vis: move.w#$0f00,$dff180;L
EMEZ KINT VAGY BENT?
     btst#5,$bfe001
     bne.svl5
     move.w#$0000,$dff180
     move.w#$4000,$dff024
     move.w#$8500,$dff09e
     move.w#$4489,$dff07e
kir: btst#5,$bfe001
     bne.skir
     move.w#$4000,$dff024
     move.l#trac,$dff020
     move.w#$9f40,$dff024
     move.w#$9f40,$dff024
     move.w#$0002,$dff09c
qw: move.w$dff01e,d2
     btst#1,d2
     beq.sqw
     btst#5,$bfe001;
     bnedisk
end: move.w#$4000,$dff024
     move.w#$0400,$dff09e
     rts

var: move.l#$e100,d7
v: sub.l#1,d7
    bnev
    rts
co: clr.ld1;PUFFER
VISZAKONV.
     move.w#$4489,trac-2
     leatrac-2,a0
     clr.ld7
vid: cmp.w#$4489,(a0)+
     bne.svid
     cmp.w#$4489,(a0)
     bne.sikp
     add.l#2,a0
lkp: clr.ld0
     move.b2(a0),d0
     move.b6(a0),d1
     and.b#$55,d0
     and.b#$55,d1
     lsl.b#1,d0
     or.bd1,d0
     mulu#512,d0
     add.ld6,d0
     move.ld0,a2
     add.l#56,a0
     move.la0,a1
     add.l#512,a1
     move.w#$100,d3
eles:move.w(a0)+,d0
     move.w(a1)+,d1
     and.w#$5555,d0
     and.w#$5555,d1
     lsl.w#1,d0
     or.wd1,d0
     cmp.b#12,d7
     bne.skopa
     cmp.b#2,d3
     beq.shalal
kopa:move.wd0,(a2)+
     dbfd3,eles
     add.l#514,a0
     addq.b#1,d7
     cmp.b#13,d7
     blt.svid
halal:rts

```

ON DISK!

Sculpt 4D

Cikk: Jazz

Lapzárta után érkezett...

Olvasóink levelek tömkelegével árasztottak el, melyből megtudtuk, hogy a ray-tracing rovat óriási népszerűségnek örvend. Ezekből a levelekből az is kiderült, hogy már mindenki ég a vágytól, hogy végre használjon (azaz hogy használni tudjon) egy ray-tracing programot. Mivel számunkra a legfontosabb az, amit ti akartok, nos kedves olvasó, örömmel jelenthetem be, hogy itt van a ray-tracing sorozat harmadik részének második cikke: A Sculpt-4D használta. Mint azt már az Amiga Magazin tavaly decemberi számában megírtam, itthon és a világon mindenhol, a Sculpt-4D a legnépszerűbb ray-tracing program. Most és a Magazin következő számaiban ennek a programnak a használatáról fogok részletesen írni.

A Sculpt-4D programot a Workbenchből indíthatod el, a program ikonjának segítségével. Némi várakozás után a drive elcsendesedik, és a gép készen áll a munkára. A képernyőn három ablakot találhatunk, ezeket hívjuk Tri-View-nak. A három ablak a munkánkhoz használt 3D teret mutatja elől, oldal, és felülnézetből. Mindhárom ablak oldalain különböző gombok (un. gadgetek) találhatók. Ezek első látásra nagyon bonyolultnak (esetleg rémisztőnek) tűnhetnek, de nem kell megijedni, bár ezek használata első látásra bonyolultnak tűnhet, mire a cikk végére minegyikkel megismerkedünk. A gadgeteket az óramutató járásának irányában fogom bemutatni, kezdve az ablak bal felső sarkában található forgató gadgetekkel, rögtön azután hogy elmondondtam az egér használatát. Az egér segítségével mozgathatjuk a mutatót, ha nem nyomjuk le az egérgombokat. A bal egérgomb megnyomására a gép a kurzort a mutató alá mozgatja. Ha a gombot nyomva tartod miközben az egeret mozgatod, úgy a kurzort mozgatod. Ha a bal gomb elengedése előtt a jobb gombot is megnyomod, úgy az dott kurzorpozícióba leraksz egy pontot. Amennyiben a kurzor egy ponthoz közel áll, miközben duplát klikkelsz a bal egérgommbal, úgy a kurzor az adott pontra áll, és a pont kiválasztott állapotba kerül (ha a pont már kiválasztott állapotban van, úgy visszaáll eredeti állapotába). A Sculpt-4D a pontokat lila, a kiválasztott pontokat feltűnő citromsárga színnel jelzi. Ha a kurzor közelében nincs ilyen pont úgy azt a program a képernyő felvillantásával jelzi, és a kurzort a helyén hagyja. két forgató gadget a kiválasztott pontokat a kurzor körül forgatja, a gadgetnek megfelelő irányba. Ha nem nyomod le egyik Amiga gombot se, úgy a a pontok 5 fokkal, a bal Amiga gommbal együtt használva 1, a jobb Amiga gommbal 45

fokkal fordulnak el. A következő gadget, és másik három társa, melyek az ablak négy oldalának közepén találhatók, segítségükkel az általuk mutatott irányba mozoghatunk a 3D térben. Itt is használhatjuk a bal és jobb Amiga gombot a hatás növelésének vagy csökkentésének érdekében. A következő két gadget, melyek az ablak jobb felső sarkában találhatók, az adott ablakot a másik elé vagy mögé helyezik az Amiga szokásoknak megfelelően. Ezek alatt találhatjuk a megfordító gadgetet. Normális állapotban az első ablak lefelé néz, de a megfordító gadget megnyomása után felfelé néz (ezt a program az ablak tartalmának tükrözésével éri el.) Ez alatt található ismét egy mozgatást végző gadget, majd ez alatt a széthúzó gadget, mely az adott ablak méretét megnöveli. Az ablak bal alsó sarkában van a méretező gadget, ami az Amiga szokásoknak megfelelően az ablak méretét változtatja meg. Ettől nem messze balra van az összehúzó gadget, ami értelemszerűen összehúzza az ablakot. Az alsó keret közepén ismét egy mozgó gadgetet találunk, majd ettől balra a középpontba állító gadgetet. Ez az ablakot úgy mozgatja, hogy a kurzor az ablak közepére kerüljön. A bal alsó sarokban van a markoló gadget, amit ki és be lehet kapcsolni. Bekapcsolt állapotában a kurzor mozgásával az éppen kiválasztott (felmarkolt) pontokat is mozgatjuk. Amikor a markoló be van kapcsolva, a kurzor is egy markoló alakot vesz fel, hogy tudjuk, hogy mivel dolgozunk. E felett találjuk az élkészítő gadgetet, amit akkor használhatunk, ha két vagy három pont van kiválasztva, és ekkor a program az adott pontokat éllel köti össze. Felette egy már ismerős gadgetet találunk, ami a mozgatást végzi. El is jutottunk az utolsó, a kiválasztó gadgethez. Ha az aktuális tri-view ablakban kiválasztottunk pontokat, úgy ezek visszatérnek normális állapotukba. Ha nincs egy kiválasztott pont sem, úgy az ablakban látható minden pont kiválasztódik. A gadget hatása nem terjed ki az ablakon kívül található pontokra. Ezzel végére is értem a gadgetek bemutatásának. Ez alapján már minden kezdő hozzáláthat a program kipróbálásához, jövő hónapban a menükről írok részletesebben, és valami értelmeset is készítünk a programmal.

Sajnos itt be kell fejeznem, ha azt akarom, hogy a lapot küldhessük a nyomdába, és idejében eljusson hozzád.

Jazz

Ray-Tracing

Itt a február, és természetesen a ray-tracing rovat várva várt folytatása. Remélem mindenkinek jól telt a hónapja, már amennyire manapság ez lehetséges. Én személy szerint amikor nem az Amigához voltam láncolva vastag, nehéz, csörgő acélláncokkal, akkor a TV-nél dolgoztam néhány animáción, az iskolában "tanultam", és félévkor angolból 4-es lettem annak ellenére, hogy felsőfokú nyelvvizsgám van (hiába, Guszti egy jó tanár). Itt az öbölháború, de remélem ez senkinek sem veszi el a kedvét egy kis ray-tracingtól. Térjünk is át valami "tényesebb" témára! Már az előző számban is érezhető volt a rovat hármasszámát: hardver, szoftver, és a ray-tracing elmélet. Így hát ettől a hónaptól kezdve a rovat három alrovatot tartalmaz. Ez a későbbiekben változni fog, mert például a következő számtól kezdve ray-tracing szoftverek használatával is fogok foglalkozni, és több más tervem is van (esetleg külön rovat az Öböl-helyzetről, ha sokáig húzodna). Mindenesetre remélem, hogy az új felosztás csak javítani fog a rovat áttekinthetőségén. Kezdjük akkor a hardverrel...

HARDVER

Az Impulse cég piacra dobott egy új 24 bites grafikus kártyát, a Firecracker 24 -et. A cég nemrég megjelent Imagine nevű programja (melyről már írtam az AM előző számaiban) közvetlenül kezeli a kártyát, így a program által elkészített 24 bites képeket a kártya azonnal megjeleníti. A megjelenítési megoldás érdekes: a kártya -egy genlockhoz hasonlóan- a 24-bites képet a normál Amiga playfieldek mögé helyezi mint harmadik playfield. Ez állítólag egyszerűbbé teszi a fejlesztőknek, hogy programjukat a kártyával kompatibilisre írják. A megjelenés időpontja nem véletlen: az Impulse cég a profi felhasználóknak az Imagine -t a kártyával együtt próbálja eladni, mint egy teljes, Amigán alapuló, profi 3D workstationt (munkaállomást). Az egy 50MHz-es GVP kártya, 8 MB 32 bites memória, és egy VTR controller segítségével olyan rendszer alakítható ki, mely versenyképes a Silicon Graphics, Sun, HP és más gyártók által készített profi 3D workstation -ökkel, viszont az ára csak töredéke ezeknek. A Firecracker 24 kártya

ára \$500 körül van, és pillanatnyilag csak az NTSC verzió kapható, ami egyelőre kizárja az európai felhasználókat.

SZOFTVER



Hihetetlen gyorsasággal terjedt el itthon a Real 3D nevű program, amely annyira új, hogy még a külföldi sajtó is csak elvétve találkozott vele. A program primitívekkel dolgozik, például: gömb, henger, téglalap, háromszög, kör, kúp. Sok lehetőség van bonyolultabb tárgyak építésére is: megnyújtás, forgatás, stb. Ugyancsak bonyolult tárgyak készítését teszi lehetővé egy csövkészítő eszköz. A tárgyak egymásból kivághatók, és sok más műveletet is végezhető velük. Az editor a Sculpt programokhoz hasonlóan a három nézeti képet mutatja, azonban a Sculpt -nál gyorsabban rajzol, és könnyebben kezelhető. Minden tárgynak megadhatjuk a színet, átlátszóságát, optikai sűrűségét, és bármilyen IFF képet is rakhatunk a tárgyra. Amikor a tárgyak tervezése kész, egy másik editor segítségével beállíthatjuk a kamera helyzetét, vagy akár egy animáció képkockáit is. A rajzolás itt is gyors, így szinte azonnal látható, hogy a tárgyak hol helyezkednek el. Ezután térhetünk át a program harmadik részébe, ami a ray-tracinget végzi. A három editor menüpont segítségével választható ki, így könnyen mozoghatunk közöttük (nem kell őket külön betölteni). Az elkészült képek minősége jó, bár szerintem elmarad a Sculpt 4D által készített képek mögött. Az IFF textúrák használatával azonban olyan képek is készíthetők, melyek messze felülmúlják a Sculpt képeket. A program néhány primitívet nagyon gyorsan rajzol ki (egy-két perc), de bizonyos esetekben nagyon lassú is lehet (Sculptét elérő vagy meghaladó idő).

Ray-Tracing

A RAY-TRACING ELMÉLETE

Most folytatom az AM előző számában megkezdett sorozatot a ray-tracing elméletéről. Legutóbb addig jutottam el, hogy leírtam annak az algoritmusnak a lényegét, aminek segítségével meghatározható, hogy létezik-e a fénysugárnak metszéspontja az adatbázisban található elemek valamelyikével. Tehát most már tudjuk, hogy fénysugár melyik elemet metszi, ezért sor kerülhet a metszéspont meghatározására. Ez annyit jelent, hogy egy térbeli egyenes és egy -három pontjával megadott- sík metszését kell megtalálni (facetek használata esetén), vagy egy egyenes és gömb közös pontját. Ennek a feladatnak matematikájával itt nem foglalkozom, akit érdekel, próbálja megkeresni a megoldást. Így már birtokunkban van annak a 3D pontnak a koordinátája, ahonnan a fény az adott képernyőpontba érkezik, és azt is tudjuk, hogy a 3D pont melyik primitív része.

A további munkához szükséges meghatározni a dőféspontban a felszín normálvektorát. Ha 3D testek a primitívek, úgy külön algoritmus szükséges minden primitívhez. Ha síklapokkal dolgozik a program, akkor az adott síklap minden pontján megegyezik a normálvektor, ami ugyanaz, mint a háromszög csúcspontjai által meghatározott sík normálvektorának a meghatározása. Az algoritmus hátránya, hogy mivel a síklap minden pontján megegyezik a normálvektor iránya, ezért a normálvektor irányán alapuló árnyékolás az egész lapot egyszínűnek fogja venni. Ez pontosan megfelel például egy téglatestnél, de egy síklapokkal közelített gömb már nem nézne ki jól, mert ennek felszíne a valóságban nem síklapokkal határolt. Ezen több módon is lehet segíteni: több síklapot használunk a tárgy szimulálásához, vagy egy "simító" algoritmust használunk. Érdekesebb ez utóbbit választani, mert a több síklap használata nagyon megnöveli az adatbázis méretét, és lassítja a program

sebességét. A "simító" eljárásról (vagy smoothing -ról) a későbbiekben fogok írni.

Ezzel kész is van a ray-tracing első alegysége, melynek bemenő paraméterei egy fénysugár (egyenes), és az adatbázis, számítási eredményei pedig a 3D dőféspont koordinátája, a dőféspontot tartalmazó primitív, és ebben a pontban a felszín normálvektora. A második alegység ezen paraméterek alapján meghatározza az adott képernyőpontba jutó fénysugár színt. Ha az elkészült képnek fekete-fehérnek kell lennie (vagyis a szürke különböző fényerősségű árnyalatokat használja), úgy csak egy intenzitás értéket kell megállapítani. Ha viszont színes képet szándékozunk készíteni, úgy három ray-tracing -et kell végezni, külön a piros, a zöld, és a kék fényre (he, he milyen jó szóvicc). Az így megkapott R, G, és B intenzitásértékek együtt alkotják a fény színt.

Ahhoz hogy a fénysugár színt meghatározzuk, meg kell tudni, hogy a dőféspontban milyen fényforrásokból származó fény hat, és ezek milyen intenzitásúak amikor elhagyják a felszínt. Ha mindet összeadjuk, akkor megkapjuk, hogy a szem az adott pontot milyen színűnek látja. A fény az alábbi forrásokból származhat: háttérfény (background vagy ambient light), lámpákból származó pontfény (lamp vagy point light), áteresztett fény (transmitted light), és visszavert fény, ami lehet teljes (reflected light), vagy fénypont visszaverődés (specular highlight). ("kerek csokoládé, szögletes csokoládé, lyukas csokoládé, csokoládés csokoládé...", ahogy egyes madarak mondanák). Gyakran egy tárgynál a különböző fénytípusok együtt lépnek fel: egy üveggömb áteresztja a fényt, de a felszínére világító lámpák fényét vissza is tükrözi. Vizsgáljuk most meg a fénytípusokat részletesen!

A háttérfény egy olyan fény, amely nem jön semmilyen határozott irányból. A valóságban úgy találod meg ezt a háttérfényt, hogy kimész az udvarra, és megnézed, hogy az árnyékok milyen sötét. Ha süt a Nap, akkor általában ez az árnyék jóval sötétebb, mint a Nap által megvilágított terület, azonban nem teljesen fekete. Ez azért van, mert valamennyi fény eljut a Napból az árnyékos területre, de nem közvetlen úton (azt te éppen elállod), hanem a levegőben szétszóródva, és a környezetben lévő tárgyakról visszaverődve.

Ray-Tracing

Ha felhős az idő, úgy lehet, hogy nincs árnyékod. Ez azért van, mert a teljes fény szétszóródik a felhőkön, így az egész fény háttérfény lesz. Mivel ez a háttérfény nem jön egy határozott irányból sem, ezért nem tudod az útját elállni, azaz nincs árnyék. Éjszaka, ha csak egy lámpa világít, úgy az árnyékod majdem vagy teljesen fekete, mivel ebben az esetben nincs (vagy nagyon kevés) a háttérfény, ami megvilágítsa.

A pontfény a háttérfénnyel ellentétben nem minden irányból jön, hanem egy pontszerűen meghatározott fényforrásból. Ezt a típusú fényforrást nagyon könnyen kipróbálhatod, csak a legközelebbi villanyt kell bekapcsolnod. Erre a fényre jellemző, hogy a távolság növekedésével az egységnyi területre jutó fényerő négyzetesen csökken, az útjába állított tárgyak árnyékot vetnek, és sima felszíneken visszatükröződik (ez a visszatükröződés a specular highlight). A pontfény érdes felületet (tehát nem tükröt) úgy világít meg, hogy a beesési pontban a felszín világossága a beesési merőleges (felszín normálvektora az adott pontban) és a beeső fény sugar által bezárt szög (a beesési szög) koszinuszával egyenesen arányos. Ezt bizonyos programok diffuse reflection -nek, azaz diffúz visszaverődésnek hívják. Sima műanyag, fém, vagy egyéb tárgyak esetében a felszín visszatükrözőzheti az az megvilágító lámpa fényét. Ez a jelenség a fénypont visszaverődés (specular reflection), melyet a későbbiekben tárgyalok.

Az áteresztett fény úgy keletkezik, hogy egy tárgy bizonyos mértékig átlátszó, és átengedi a fényt. Az előző fénytípusokkal ellentétben itt nem tudjuk közvetlenül kiszámolni, hogy a fény milyen erős, mert nem tudjuk, hogy a tárgyon milyen fény sugar halad keresztül. Ezért szükséges egy új fény sugar indítása. Ez a fény sugar nem egyezik meg az éppen vizsgálttal, mert amikor a fény az egyik közegből a másikba lép, akkor megtörik. Ez azt jelenti, hogy a beesési szög (a) nem egyezik meg az új fény sugar és a beesési merőleges által bezárt szöggel (b). A két szögre az alábbi állítás igaz: $\sin(a) / \sin(b) = n_2/n_1$. Az n_2/n_1 a két közeg által meghatározott konstans, a törésmutató.

Néhány anyag törésmutatója levegőre vonatkoztatva:

Levegő 1.000
Jég 1.310

Víz 1.333
Üveg 1.515 - 1.751
Borostyán 1.540
Gyémánt 2.417

Az új fény sugarat paraméterként használva újra meghívjuk (esetleg rekurzívan) a ray-tracing funkciót, és a funkció által visszaadott fényértékekkel fogunk tovább számolni. Innen látszik, hogy miért olyan lassú a ray-tracing üveg tárgyak esetében: minden fény sugarhoz tartozik egy új, áteresztett fény sugar, amivel tovább kell számolni, tehát a számítási idő minimum kétszeresére nő. Az is valószínű, hogy a fény sugar ugyan így ki is fog lépni a tárgyból, ezzel is egy újabb, tört fény sugarat állítva elő, ami ismét tovább növeli a számítási időt.

Bizonyos felszínnek olyan simák, hogy a rájuk eső fény nagy részét visszatükrözik. Ha egy ilyen tárgyat nézünk, akkor annak a felszínén a környezetet látjuk visszatükröződni. A jelenség könnyen kipróbálható bármilyen, a kezünk ügyébe eső tükörrel. Miután részletesen megvizsgáltuk a jelenséget, (és megelégedetten tudomásul vettük, hogy milyen szépek vagyunk), nézzük meg a fényvisszaverődés tulajdonságait! A fénytöréshez hasonlóan a fényvisszaverődésnél is értelmezzük a beeső fény sugarat, a beesési merőleget, és a beesési szöget. A felszínről visszaverődő fény sugarat visszavert fény sugarnak (reflected light) hívjuk, a beesési merőleges és a visszavert fény sugar által bezárt szöget pedig -értelemszerűen- visszaverődési szögnek. A fényvisszaverődésre igaz a következő: beesési szög = visszaverődési szög. Mivel ismerjük a beeső fény sugarat (ez éppen az általunk vizsgált fény sugar) és a beesési merőleget (ez a felszín normálvektora), a visszavert fény sugar könnyen megtalálható.

Létezik egy másfajta visszaverődés is, a részleges vagy fényfolt visszaverődés. Ebben az esetben a tárgy felszínén nem a más tárgyakról ráeső fény sugarakat tükrözi vissza, hanem egyenesen a lámpa fényét (az ilyen, lámpa által okozott fényfoltot hívjuk specular highlight -nak). Ez a jelenség nem csak a szigorúan vett tükörök felszínén észlelhető, hanem minden, kellően sima felszínnű tárgyon. Akkor lép fel, ha a visszavert fény sugar és a lámpából a beesési pontba húzott egyenes által bezárt szög kisebb egy bizonyos értéknél (a Phong

Ray-Tracing

spectral reflection coefficientnél). Minél simább egy tárgy felszíne, annál kisebb a fényfolt, és annál kisebb az a beesési szög, ami alatt a felszín visszaveri a lámpa fényét. Fémes tárgyakról visszavert fény a fém színével, míg nemfémes sima felszínnekél a lámpa színével egyezik meg.

Miután megvizsgáltuk a különböző fénytípusokat, most már összerakhatjuk őket egyetlen algoritmussá, amely meghatározza a képernyő adott pontján áthaladó fénysugár színet. A program összeköti a beesési pontot a lámpával, majd megnézi (a már leírt módon), hogy a kettő között van-e bármilyen tárgy. Ha igen, akkor a lámpa fénye nem jut el a vizsgált pontba, így azzal nem kell számolni. Ha a lámpa útjában nincs akadály, akkor az megvilágítja a felszínt. A beesési pontban a fényerősséget úgy számíthatjuk ki, hogy a lámpa-beesési pont távolság négyzetének arányában a lámpa fényerejét csökkentjük. Mivel színes ray-tracingben dolgozunk, a lámpa R, G, és B fényértékére ezt külön-külön ki kell számolni. Az így megkapott érték azt mutatja meg, hogy a lámpából mennyi fény jut a beesési pontba. A fényt a beesési szög koszinuszának arányában csökkentjük, és így megkapjuk az adott lámpához tartozó, a tárgy által visszavert fényt (diffuse reflection). Ha a tárgy sima, úgy fényfoltot is kell számolnunk. Összehasonlítjuk a visszavert szöveget a lámpa fénysugara és a beesési merőleges által bezárt szöggel, és ha ez kisebb, mint a Phong spectral reflection coefficient, úgy a felszín adott pontján fényfolt van (specular highlight). Ennek fényességét egy, a felszínen jellemző konstanssal szorozzuk, majd az eredeti lámpafény színét eltoljuk a tárgy színének irányába, a fémességi konstansnak megfelelően. Ezt minden lámpával megcsináljuk, és a kapott RGB fényértékeket összeadjuk.

Ha a tárgy áteresztja a fényt, úgy ki kell számolni, hogy mennyi az áteresztett fény értéke (refracted light), és ezt az előzőekben kiszámított összeghez kell adni. Ezt meg kell még csinálni a tükröződésre is (reflected light), ugyancsak hozzáadva az előző összeghez. Hátra van még a háttérfény hozzáadása, és már meg is állapítottuk, hogy összesen mennyi fény jut a beesési pontba. Ennek a fénynek egy részét a tárgy elnyeli. Ezek az R, G, és B elnyelési értékek határozzák meg a tárgy színét. Például, ha egy tárgy piros, az azt jelenti, hogy a zöld és kék fényt elnyeli, és csak a pirosat veri

vissza. Az algoritmusunk végső eredménye három szám, melyek meghatározzák, hogy a felszín a beesési pontban milyen színű, azaz a szembe az adott képernyőponton keresztül milyen intenzitású RGB fény jut.

Ezzel a ray-tracing be is fejeződik, most már csak a képpontok megjelenítése van hátra. Ha ray-tracerünk 24 bites képet készít, úgy az RGB világossági értékek mindegyikét 8-bitesre alakítja. Érdekes a ray-tracer-be beleépíteni, hogy ki tudja ezt a képet menteni, mert ezt egy 24-bites frame-buffer segítségével megjeleníthetjük, vagy egy másik, általunk készített formátumra hozhatjuk, és tovább dolgozhatunk vele. Ha a normál Amigán akarjuk a képet megjeleníteni, úgy az alábbi formátumok valamelyikébe érdemes leképezni: HAM, Super-HAM, vagy Dynamic HiRes. Pillanatnyilag a forgalomban lévő ray-tracing programok csak HAM-ba tudnak leképezni. Ennek módjáról itt nem írok, a lényeg az hogy egy 24-bit - HAM konvertert kell készíteni. A Super-HAM egy olyan HAM megjelenítési mód, ahol a színregiszterek minden raszter-sorban újra vannak írva, így a HAM-nél jobb minőségű, élesebb kép készíthető. Az AM titkos és bizalmas értesülései szerint már vannak, akik írtak 24-bit - Super-HAM konvertert ray-tracinghez itthon Magyarországon (Kosír Attilának), és ezt sikerrel használták. A Dynamic HiRes egy megjelenítési mód, ahol 640x512-es felbontással lehet 4096 színű képeket előállítani. Pillanatnyilag csak a DigiView képdigitalizáló legújabb verziója kezeli, ray-tracing felhasználásról még nem hallottam. Játékprogramokhoz lehet érdekes egy olyan leképző készítése, ami 16 vagy 32 színű képfarmátumban készít képeket, mert az így megrajzolt tárgyak mint spriteok vagy BOBok használhatók.

Az eddigi rutinokat összerakva már kész ray-tracing programunk van. Néhány profi eljárásról, mint kód szimuláció, simítás, antialiasing, texture- és bump-mapping a következő számban lesz szó. Azoknak, akik nem akarnak maguk ray-tracing programot írni, csak egy már meglévő programot használni, azoknak se kell elkeseredni: a következő számtól kezdődően nekilátunk a ray-tracing programok használatának, kezdve a Sculpt 4D-vel. Addig is, jó munkát, szórakozást (kinek mi), és a Fény legyen veletek!

Jazz

Desktop Video: Deluxe Paint



Jogos kritikát kaptunk, miszerint sokmindent ígérünk, amit nem valósítunk meg. Ezen meg nem valósított ígéretek egyikét most azonban valóra váltjuk: közölni fogjuk a Deluxe Paint III program leírását. Erre nagy igény van, hiszen ez a program a TOP listánk élén van, mindenki használja. A Deluxe Paint-re jellemző adat, hogy majdnem annyit adtak el belőle 1988-ban az USA-ban, mint amennyi A500-at. (Ezt próbálják meg Magyarországon...). Fölvetődhet a kérdés, hogy minek erről a pofon egyszerű, logikusan felépített programról leírást írni? A válasz, hogy nem egyszerűen csak leírást fogunk közölni, hanem egy rovatot szeretnénk indítani a Desktop Video-n belül a Dpaint számára. Ez a program ugyanis nem más, mint egy munkaeszköz. Az ecsetet is pofon egyszerű, és logikus használni, mégis számtalan könyv jelent már meg a művészetről, de akár a "pointilista ecsetkezelés rejtelmeiről" is. A kérdés tehát, hogy mi lesz ebben a rovatban:

- A program leírása: a szerszámok leírása, az egyes funkciók leírása menüpontoként részletezése, rajzolómódok, szerszámok, stb., memóriakezelés.

- Trükkök, technikák: mivel ezt az eszközt mindnki másképpen használja, ezernyi munkamódszer, apró fogás létezik, amit az ember megtanulhat. Ezek közül van nekünk is néhány a tarsolyunkban, néhányat lefordíthatunk a nyugati szakirodalomból, de elsősorban azt szeretnénk, ha azok, akik szintén sokat dolgoznak a DPAINT-tal, azok elküldenék a saját trükkjeiket, technikáikat.

- gyors munkához: billentyűzet-rövidítések

- Animáció a DPAINT III-mal. A profi animáció munkamódszerétől kezdve, a program által használt delta fájlokból álló animációs formátum részletes ismertetése, a szükséges összes PD program mellékelésével, és leírásával együtt.

A Deluxe Paint-et az Electronic Arts hatalmas stábja készítette, mintegy 12 ember dolgozott rajta, hogy a leglogikusabb koncepciók alapján működjön. A projekt mozgatórugója DAN SILVA volt, aki maga is "számítógépes művész", egyike azoknak, akik leghamarabb felfedezték a számítógépet, mint a művészet egyik hatékony médiáját. Természetesen a dolog úgy kezdődött, hogy megjelent a piacon az első igazán szuperszonikus, de elérhető árú számítógép, az AMIGA, ami annyira megmozgatta a fantáziáját, hogy azonnal nekilátott, a DPAINT tervezéséhez. Mire az AMIGA 500-as piacra került, ő már kész volt a Deluxe Paint nevű programmal, ami hatalmas sikert aratott. Sokan, mint most a Toaster, és a 2000-es esetében, akkor csak azért vették az 500-ast, hogy futtatni tudják az olcsó "Paintbox-ot". Ők azonban nem ültek sokáig a babérjaikon, hamarosan elkészültek a továbbfejlesztett II-es, majd nemrégiben az animációs lehetőségeket is magába foglaló III-as verzióval is.

Jelenleg a legfejlettebb verzió a 3.25-ös, amiről mi tudunk. Ez nem bízva a véletlenre az NTSC-PAL gépek kezelésének a különbségét: a bejelentkezéskor a SCREEN FORMAT requester-ben kiválaszthatod, hogy a PAL, illetve NTSC a géped. Más látványos változtatás nincs benne a III-as verzióhoz képest. A memóriakezelése jobbnak tűnik, bár ezt őszintén szólva nem teszteltem "tudományosan". Egy azonban biztos: a III-as verzióban van egy csomó BUG, amit a 3.25-ben már nem találtam meg. Például ha 1M Chip RAM-nál kevesebb van a gépedben, és egy nagyobb darab ANIM BRUSH-t próbálsz meg Z tengely körül forgatva animálni, a program az első néhány lépés után elszáll. Arról nem is beszélve, ha MOVE-val megadsz egy mozgatót, gyakran előfordul, hogy a végén nem marad elég memóriád. Ilyenkor egy "Not enough memory to save changes" hibaüzenetet kapsz. Itt hiába klikkelsz a "proceed" gadget-re, a MOVE tovább működik, össze-vissza zagyválva az addigi munkádat. Alapvetően igaz, hogy a III-as verzióban, ha bármi miatt a memóriád teljes kihasználásának a szélére kerülsz, az biztos, hogy előbb utóbb a jól ismert "Software failure, Task Held, Finish ALL...", illetve a villógó vörös keret fogod látni. Ezeket a hibákat a 3.25-ösnél nem tapasztaltam. Mindenesetre, az alapvető munkamódszer, és elsősorban szabály a DPAINT használatakor is, hogy ha készen vagy a munkád egy fontos fázisával, mielőtt valami olyat csinálsz, amiben nem vagy 100%-ig biztos, hogy

Desktop Video: Deluxe Paint

rajzolni, mint a grafitceruzával. Ha készen vagy az arányok felvételével, akkor rajzold meg a végleges körvonalat a folyamatos rajzoló funkcióval.

3. Folyamatos rajzoló funkció: (continuous freehand tool)

Ezzel a funkcióval mindig folyamatos vonalat húzol. Ha nagyon gyorsan húzod az egeret, akkor nem szünetek keletkeznek, hanem a görbe nem pontosan fogja követni a kezéd mozgását, szögletes lesz. Ebben a módban tulajdonképpen a szüneteket, amiket a pontozott módban hagy, összeköti egy folyamatos vonallal. Figyeld meg, hogy minél kisebb az ecset, annál gyorsabban tudsz vele rajzolni. Az ezzel a funkcióval rajzolt görbéket beontheted festékkel, nem fog kifolyni, hiszen a görbe folyamatos.

4. Egyenes vonalrajzoló funkció: (straight line tool)

A kiválasztásnál szintén nincs beállító szerepe a jobb egérgombnak. Mozdasd a hajszátkeresztet a rajzolandó vonal egyik végpontjára, majd nyomd le az egérgombot, és tartsd is nyomva. Húzd ki a vonal másik végpontját, és engedd el az egérgombot. A bal gombbal a rajzolószínnel tudsz rajzolni, a jobbal a háttérszínnel. Természetesen a vonalat az éppen aktuális ecsettel rajzolja meg, ami lehet felvett ecset is.

5. Görbe rajzoló funkció: (Curve tool)

Ez egy nagyon "póverful" funkció. Hasonlóan működik, mint a vonalrajzoló, de a végén még egy klikket el kell követni az eredmény eléréséhez. Miután megadtad a görbe két végpontját (ugyanúgy mint a vonalrajzolóval), mozdasd tovább az egeret. Ehhez a harmadik ponthoz fog simulni a görbe. Matematikus agyúak tudják, hogy mit jelenet az, hogy egy görbe egy ponthoz simul, és rögtön jól tudják használni ezt a szerszámat, de azért nem árt egy kicsit gyakorolni.

Ezzel szerszámmal pillanatok alatt rajzolhatsz szabályos görbéket. Például egy sematikus szív négy ilyen szabályos görbéből áll. A szív bal oldalát megrajzolhatod két görbéből, majd ezt felveszed ecsetként, Y tengelye körül megfordítod (x billentyű), és ezt hozzáragasztod a már megrajzolt görbéhez. Ezzel kész is a szabályos szív körvonala, amivel egyébként fél óráig kellene szenvedned, ha szépen meg akarnád rajzolni.

- folytatjuk -

24 bit a közelben !

Igazán nem akarom Jazz kenyerét elvenni, de mivel ő elsősorban a távoli amerikai helyzetet ismeri, megpróbáltam utánnanézni, hogy az általa írt hardvercsodák hol, és mennyiért kaphatók a "pocsolya ezen a felén".

24 bites kártyák közül a következőket találtam azonnal kaphatónak:

- COLORBURST

A COLORBURST 16.8 millió színű IFF képek megjelenítésére alkalmas 768*580-as felbontásban. Természetesen működik a SCULPT 4D, TURBO SILVER, és az IMAGINE-nel is. A Digiview Gold képes 24 bites képeket digitalizálni, amiket a MEGAPAINTEVŰ 24 bites Paint-programmal változtatgatunk. A COLORBURST minden GENLOCK-kal kompatibilis. Ez a kártya GRAZ-ban 7099 ATS-be kerül. Az ANUBIS KFT vállalja a behozatalát, ha megrendeled tőlük.



-TARGA, és VISTA kártyák

Professzionális kártyák, amelyek kiépítéstől, és típustól függően 4270-9640 DM-be kerülnek. Ezek nem kimondottan Amiga perifériák, ugyanezek a kártyák működnek gyakorlatilag az összes SUN, és egyéb IBM AT alapú munkaállomásban, például a Novotrade animációs stúdiójában. Egy Amiga 2000, és egy AT bridgeboard segítségével tehát akár az eredeti VISTA animációs programot is futtathatjuk, igaz ez a megoldás már semmivel sem olcsóbb (csak jobb), mintha egy 486-ost használnánk a rendszer alapjául.

Amennyiben igény lesz rá, közöljük a rendelkezésünkre álló összes információt a COLORBURST-ről, és a többi 24 bites kártyáról egy cikkben.

BB KING

ZENE

Kezdjük mondjuk azzal, hogy sietve leszögezzük: a továbbiakban határozott különbséget igyekszünk majd tartani két, egymástól eléggé elkülöníthető kategória között. Bár mindkettő bőven belefér a 'számítógépes zene' kategóriájába, egészen másképp közelítendő meg az Amiga hardware által megszólaltatott zene, illetőleg az, amikor a gépet kimondottan midi-adatok rögzítésére és célelvű (jó szó, mi?) manipulálására használjuk fel. Természetesen számos átfedéssel találkozhatunk, például több profi midi-program ad lehetőséget a beépített hanggenerátor megszólaltatására (és nemcsak metronómként), valamint minden magára valamit is adó soundtracker-verzió és egyéb amigás zeneszerkesztő software képes kezelni midi-információkat is, ez azonban a lényegen nem változtat. Való igaz, hogy mindkét esetben zene készítése a cél (mint ahogyan az élő zene elvakult hívei általában fájdalmas nyüszítéssel panaszozik fel, mennyire 'csinált' is az ilyen zene [az efféle idealisták etetése és ingerlése szigorúan tilos! -a szerk.]), a rendelkezésre álló eszközök és a tanácsolható módszerek között azonban rendszerint éktelen különbségek vannak. A jövőben tehát rovatunkban is igyekszünk kettéválasztani ezeket a területeket, és lehetőség szerint külön foglalkozni velük.

I. ZENE AZ AMIGÁN

Különbféle színvonalas és kevésbé színvonalas kiadványokban már rengeteg leírást olvashattunk arról, hogyan is kell kezelni mondjuk egy soundtrackert. Végeredményben ez utóbbi könyvelhető el úgy, mint a definitív (azért ez se rossz szó) zeneszerkesztő imádott gépünkön. Időszzerűnek látszik tehát egyszer végre megkísérlni az 'emberi oldalról' (ah!) közelíteni a dolgokhoz, amolyan Számítógépes Zeneszerzési Kiskaté Továbbá Gyorstalpaló Tanfolyam jelleggel. Avagy: mi is kell ahhoz, hogy jó zenét írjunk?

Nos, feltétlenül kell: 1. Idő (sok), 2. Amiga (legalább egy), 3. fantázia (minél több), 4. hallgatóság (minél tapintatosabb, mert annál később veszítjük el végleg az utolsó csepp lelkesedésünket is). Természetesen ez a lista roppantul elnagyolt. Fentiekén kívül még ezernyi dologra szükség lehet, de legfőképpen és leginkább: minél több használható hangmintára. Enélkül akár bele se kezdünk.

Hogyan szerezzünk be sample-eket? Ennek számtalan módja van. Írhatunk a télapónak, de ezzel a módszerrel nehéz biztosra menni. Jobb tehát, ha a család tűrőképességét a végsőig feszítve veszünk (jó közügyességgel rendelkezők esetén fabrikálunk) egy hangdigitalizálót, szűk zsebpénzen élők végső esetben vázolhatják a családnak a kimustrált cipőtálpak gasztronómiai jellegű hasznosításának vonzó, és hosszabb távon anyagilag is előnyös távlatait is. Ha meggyőzőképességünk hatott, a beszerzett hangdigitalizálóval már bátran nekivághatunk a gyakorlati hangtan fizikailag már alaposan feltárt, ám fiziológiailag annál talányosabb rejtelseinek. Az utóbbi évek zenéjében egyre inkább a hangzás illetve a 'megszólaltatás' minősége látszik dominálni. Érezhető az igény, hogy minden egyes darabnak 'másképp' kell szólnia. Márpedig ha követni kívánjuk a trendet (amire előbb vagy

utóbb úgyis rákényszerülünk), nem elégedhetünk meg néhány játéprogramokból kilopkodott hangmintával, hanem egyéni igényeinknek megfelelőkre kell szert tennünk, mégpedig minél nagyobb számban. Magától értetődően itt is érvényesül Murphy törvénye, amely esetünkben kb. így hangzik: mindig egy olyan hangmintára lenne szükségünk életünk éppen aktuális főművének megalkotásához, amilyen jusst se áll rendelkezésünkre.

Természetesen megfoghatjuk a dolognak a könnyebbik végét is, ha úgy döntünk, hogy kizárólag az innen-onnan felvett lemezeinken megtalálható mintákat használjuk, ebben az esetben azonban tekintélyes gyűjteményt célszerű összeharácsolni, ami viszont rendszerint gátlástalanságunk mértékével fordítottan arányos mennyiségű időbe tellik. Szerencsére a legújabb sorozatú soundtracker-verziók képesek az analóg hangkeltés szimulálására is, tehát pl. beállíthatunk paramétereket a frekvenciamodulációs hanggenerálást utánózva (a YAMAHA cég DX-sorozata működött ezen az elven). Ha csak meglévő mintákból kívánunk gazdálkodni, vigyázzunk azonban arra, hogy ne essünk bele a de-jól-szól című hiú megalégedés csapdájába. Ha néhány jó hangmintából felépítünk egy egyszerű pergődob-nagydob alap ritmuskíséretet, és beütögetünk mondjuk egy fémes 'ElectricBass'-menetet, az már önmagában ugyan szólhat jól, de ezzel még nem haladtunk messzire a hallhatatlanság felé vezető rögzítő úton. A bölcsek szerint még mindig a tiszta melódia az, amiről a zene és általában a zenélés szól, és végeredményben ezt az érvet igen nehéz megcáfolni, pedig azzal, hogy a modern technika gyakorlatilag bárki számára lehetővé tette a zeneszerzést és -alkotást, pusztán a nagy számok törvényével összhangban is hallunk néhány valóban szélsőséges megnyilvánulást. Ritka az az eset, hogy csak a technika képes legyen 'elvinni' egy dalt, bár erre is találunk példát. Végül is

ZENE

leggyakrabban egy-egy dallamrészlet felidéződésére támad ingerenciánk újra meghallgatni egy-egy zenedarabot, és ezalól az Amigán létrehozott zene sem lehet elvi kivétel.

Ha azonban uram bocsá' akad néhány saját elképzelésünk, előbb-utóbb érezni fogjuk a szükségét annak, hogy egyedi mintákat is készítsünk. A sampling-technika körülményessége és memóriaigényessége ellenére is óriási lehetőségeket rejt. Fent már említett bölcsek szerint a melódia szerepének háttérbe szorulása mellett a számítógépes zene másik nagy negatívuma a 'kifejezés' elsovasztása, és bizony ebben is van valami. A szó itt 'kifejező játékmód' értelemben veendő, és arra utal, hogy amíg mondjuk egy gitárosnak a pengetés módjától kezdve annak dinamikáján keresztül a húrok feszítéséig vagy a vibratóig (hangmagasság lebegtetése) ezerféle lehetőség áll a rendelkezésére az igazán kifejező játéktílus megfoghatatlan kategóriájának elérésére, addig szegény amigás komponista kénytelen beérni a soundtracker vibratójával és a még profi szintetizátorokon is körülményes portamentójával (fokozatos hangmagasság-elérés, 'hajlítás'). Továbbá csak a zeneóvodában hihetik azt a kölkök, hogy a dinamika egyenlő a hanerővel. Még egy lepusztult zongorán is észrevehető, hogy a gyengébben leütött hangok nemcsak hangerejükben, de hangszínükben is mások, mint a heves ütlegelés módszerével megszólaltatottak: a dinamika növelésével a megszólaló hang egyre több felharmonikust tartalmaz, kvázi világosabb, élesebb. Fent említett 'kifejező játékmód' pedig akusztikailag annyit tesz, hogy a fülnek nyilván érdekesebb, színesebb élményt nyújt a hangképükben többé-kevésbé különböző hangok egymásutánja. Mit jelent tehát mindez a mi esetünkben? Elsősorban azt, hogy amennyiben mégis van hangdigitalizálónk, csengessünk be gitáros barátunk ajtaján (gitáros barátja tapasztalataim szerint mindenkinek van; akinek nincs, az menjen moziba inkább), és közöljük vele, hogy ő most hangmintákat fog bejátszani a hónunk alatt cipelt Amigába. Az erőteljes fellépés igen sokat segít. Monitort ne vigyünk, nyugodtan felszólítást adhatunk a család-szeme-fénye új ITT-NOKIA televízió áttelepítésére (azokon ált. van SCART csatlakozó). Készítsünk rengeteg hangmintát, mégpedig különböző játékmódokkal. barátunk csak ne sajnálja a torzítóját és az egyéb effektjeit, és (nagyon fontos!): különböző hangmagasságokban, célszerű oktávonként vagy méginkább fél-oktávonként. Ne csak tiszta

hangokat vegyünk fel, hanem egyebeket is: 'nyávogtatás' (nem a macskáé!), hajlítások stb. Mellékesen a macskát is digitalizálhatjuk, ha elég bátrak vagyunk. Ha pedig a karhatalomnak végre sikerült barátunk lakásának elhagyására bírnia bennünket, és sikeresen hazaértünk, kezdjünk el kísérletezni a felvevő mintákkal. Próbáljunk meg több mintából létrehozni egy szólamot, a dinamika függvényében változtatva azokat. Ha ügyesek vagyunk, az Amiga egészen technikás gitárszólót fog lejátszani a végén.

Természetesen gitáros barátunk esete csak egy a lehetséges rengeteg közül. Az elv azonban ugyanaz. A küngető csincsillák mellett hasznos lehet a szürkeállományt is munkára fogni.

Ha megérjük a jövő hónapot, igyekszünk majd tovább firtatni, milyen hangmintákkal is célszerű kezdeni (tapasztaltabbak esetén folytatni) zeneszerzői karrierünket.

II. MIDI A DIVAT?

Fentebb már említett gitáros barátunkkal szemben az Amigának (és az összes számítógépnek) van egy abszolút elvitathatatlan előnye: nem beszél vissza, nem vitatkozik, és nincsenek a miénktől eltérő, zűrösen kódos elképzelései a MŰVÉSZET parnasszusi magaslatairól. Valószínűleg ez a tény vezérelhetette a hetvenes évek végén a Sequential Circuits fejlesztőmérnökeit abban, hogy egy általános szabvány bevezetését szorgalmazzák az elektronikus hangszerek világában, amelynek segítségével egy tetszés szerinti, akár különféle gyártmányú eszközökből álló rendszerben is a gépek digitális információkat küldhetnek egymásnak pillanatnyi állapotukról, illetve reagálhatnak az üzenetekre. Így született meg a midi. A zene mindig is közel állt a matematikához, pontosabban ez a művészetek közül számokkal leginkább leírható terület. Valójában professzionális téren a midinek nem az a legnagyobb előnye, hogy az egyik szintetizátoron lenyomunk egy billentyűt, és a másik szólal meg, hanem az, hogy segítségével (és a megfelelő számítógépprogramok használatával) a zene a hétköznapi ember számára is tudományos, esetünkben számítástechnikai megközelítéssel analizálható és átlátható, banálisán: a számok nyelvére lefordítható tevékenységgé vált. És persze: megszületett a 'nagyzenekar a kissozában' koncepciója. Nem a mi dolgunk eldönteni, hogy szigorúan esztétikai szempontokból mindez mennyire minősíthető előremutatónak. De hogy

ZENE

játéknak egészen kiváló, ahhoz nem férhet kétség.

Minden zenével foglalkozó lelkes ifjú, akit valaha is beoltott a modern hangzások imádatának makacs vírusa (URZ talán erre is tud ajánlani valami viruskillert), eljut odáig, hogy egy szép napon szeretne újabb kezeket növesztetni, mert két kézzel még a legelszántabbak is csak két hangszeren játszhatnak egyszerre. Mivel ez a szándék laboratóriumi beavatkozás nélkül egyelőre kevés sikerrel kecsegtet, rövid reménytelenség után legtöbbjük rátalál az ésszerűbb megoldásra, és érdeklődni kezd a midi iránt.

Félretéve a lírát, a professzionális szintű midi-programokat alapvetően két csoportba sorolhatjuk. Az egyikbe azok tartoznak, melyekkel gépünket egyfajta digitális magnetofonná alakíthatjuk, és felvehetjük mindazt, amit hangszerünkön játszunk. (Gyengébbek kedvéért: nem a hangokat vesszük fel, csak a billentyűzet tevékenységének digitális információit; hangszerünkön új hangszínt beállítva más hangzást kapunk.) A másik csoportban a számítógép segítségével a szintetizátor pillanatnyi hangszínét változtathatjuk bizonyos paraméterek alapján. A midi-protokoll e célra tartja fenn az ún. SYSTEM EXCLUSIVE üzenetet, amely gyártó- és modellspecifikus adatok küldözgetését teszi lehetővé egy egyébként szigorúan szabványos környezetben.

Midi-ügyben az Amiga nehéz helyzetben van, hiszen a professzionális zenei felhasználók körében a legteljesebb az Atari megsemmisítő fölénye, aminek egyetlen oka az Atari-gépekre létező elképesztő midi-szoftver túlkínálat (meg az a másik marha, aki kihagyta a midi-interfészt az Amigából). Egyéb hardver-okok természetesen nem támasztják ezt alá. Azért a piacorientált nyugati világ szerencséje az Amigára is megteremtett egy megbízható szoftverbázist, úgyhogy aki ezzel a géppel kívánja meghajtani hetvenhétezer szintetizátorát és modulját (billentyűzet nélküli szint), annak sem lesznek különösebb gondjai. [Plusz atarisok kezében eddig kizárólag méregdrága originált bolti floppykat láttam midi-program felkiáltással, nem így a másik tábor... khmmm..., ajaj..., ejnye... stb.]

Az amigás midi-szoftverek egyik etalonja (első csoport, lásd feljebb) a Dr T. zengzetes névre hallgató amerikai cég Keyboard Controlled Sequencer (KCS) című remekműve, amelynek jelen pillanatban legjobb változata a 3.0-ás verzió. A programnak egyébiránt Atari és

Macintosh változata is kapható, és - láss csodát - gyakorlatilag ugyanúgy működik és ugyanannyit tud (így van, és ugyanúgy néz is ki). Tervezzük, hogy, tekintettel a program bonyolultságára, a közeljövőben részletes leírást közlünk róla. Ezt azért bocsátottuk előre, hogy nyomtatékosan felhívhassuk az érdeklődők figyelmét, hogy semmiképp ne hívjanak fel bennünket, mert nem fogjuk nem megmondani nekik, hogy szükség esetén honnan nem tudják nem megszerezni ezt a programot, és nem fogjuk nem hozzátenni azt sem, hogy mennyire elítéljük a disk swapping-féle üzelmeket, és igazán nem értjük, hogy akinek kell ez a program, az ugyan miért nem ugrik be a Dr T. midi-szaktoltjába a volt Lenin, ma Botos Katalin Tárcá Nélküli Miniszter Asszony körúton.

A programnak egyébként rengeteg változata van, némelyikük eleve más névvel kerül forgalomba. A Midi Recording Studio például egy drasztikusan egyszerűsített verzió, amelyen nyolc csatornára rögzíthetünk adatokat, és egyértelműen magnót szimulál a szoftver, nincs lehetőség a rögzített zenei részeket később egyenként név alapján (versszak, refrén stb.) sorrendbe rakni. Ugyanakkor világosan átlátható az egész. Igen fontos ugyanis, hogy olyan szoftvert válasszunk, amely összetettségében azonos szinten van hangszerparkunk fejlettségével. Akinek mondjuk egy automata kíséretes Casiója van, nem tanácsolható pl. a KCS 3.0, mert csak a képernyők között való ugráندozásokkal és egyéb szükséges tevéşkedésekkel több idő fog elmenni, mint a hangszer megszólaltatásával. Ugyanakkor, ha valaki a legújabb magyar filmhez ír olyan modem szintetizátoros zenét, amilyen a legtöbb nyugati filmhez készül, (és Isten adná, hogy végre megbíz zanak valakit ilyesmivel Magyarországon), akkor csak a KCS lenne ajánlható a maga 48 sávjával, ugyanis a filmes- és videós világban használatos, ún. SMPTE szinkronjelre is képes ráállni. Tartozik továbbá hozzá egy kiegészítő program is, Copyist fedőnév alatt, mely a KCS-ben kimentett kész zenei anyagokról partitúra-jellegű kottát is nyomtat. Plusz kávéf főz, és bevásárol...

Nos, bevezetőnek ennyit szántunk. Természetesen szívesen veszünk mindenfajta javaslatot és kérdést, hiszen igen nehéz felmérni, az amigások általában milyen mélységben érdeklődnek a modern zene kulisszatitkai iránt: vajon hányan zenélnek közülünk aktívan, hányan foglalkoznak azzal, hogy zenét írjanak demókhoz és programokhoz, és hányuk olvasgatnak csak úgy általában szívesen a zene kérdéseiről.

PAGESTREAM

Kis lekiismeretfurdalás gyötör bennünket, mert a januári számban egyáltalán nem foglalkoztunk DTP-vel. Reméljük kárpótlásul jó lesz a kiéhezett Amigás DTP rajongóknak ez a sorozat, ami az egyik legjobb programról a Pagestream-ről szól majd. Ez a rendszer kipróbált, és jól működik. Compi jóvoltából létezik a Pagestream kompatibilis ékezetes vektorfont készlet, amit az újságban reklámozunk. Ez NEM Public Domain, nem szabadon terjeszthető. Csaknem két hónap munkája van benne. Olyan árat szabtunk neki, amit biztosan mindenki ki tud fizetni, aki profi módon akar DTP-vel foglalkozni: 500 Ft. A lemez szabadon másolható kipróbálható: shareware. Ez azt jelenti, hogy ha tetszik, és használod, akkor a becsületedre van bízva, hogy Compi-nak küldj 500 Ft-ot, aki két hónapig dolgozott azon, hogy ezt a programot magyar ékezetekkel tudd használni.

A Pagestream programot teljes részletességgel kívánjuk ismertetni. Ezen a programon keresztül megismerkedhetünk a DTP programok működésének alapszabályaival, így egy másik program használatát adott esetben már sokkal könnyebb lesz megtanulni.

A szükséges hardware:

A program futtatásához minimum 1MB RAM-ra van szükség. Nagyobb összefüggő dokumentumok esetén (például egy újság, vagy könyv) ez a memória azonban kevés, ilyenkor 2-4 MB-ra is szükségünk lehet. Egy meghajtóval roppant kényelmetlen a munka, a sok lemezcseré miatt, ezért ajánlatos egy külső drive-ot is beszerezni. Egy ilyen bonyolult program, mint a DTP, elég nehézkesen, lassan fut a standard Amigán, ha komolyan akarunk a dologgal foglalkozni feltétlenül szükségünk van egy turbókártyára. A Pagestream alkalmas hobby célokra is, leveleket, szórólapokat nyomtatni, akár 9-tűs nyomtatón is. A Pagestream meglehetősen szépen eredményt tud elérni egy ilyen olcsó nyomtatóval is, mint pl. a STAR LC-10-es. Az AM szórólapjai például így készültek

annak idején.

Profi munkára is alkalmas a program, képes lézernyomtatót is vezérelni, így 300 DPI-s felbontásban nyomtatni. Ez persze még nem a technológia csúcsa, hiszen a Pagestreamnek van Postscript kimenete is. Ez azt jelenti, hogy a nyomdatechnológiában használatos lézerlevilágítókat is meg tudja hajtani, amivel akár 8000 DPI-s felbontást is elérhetünk. A Pagestream tehát alkalmas a legprofibb DTP munkára is, például színes újságok készítésére is. Íme néhány ajánlott konfiguráció különböző felhasználási területekre:

Otthoni hobby használatra, iskolának, műv.házaknak: (szórólap, levél)

- AMIGA 500 (1MB RAM)
- STAR LC-10 nyomtató
- (külső drive)

Iskolaújság, kisebb újság készítése:

- AMIGA 500 (1MB RAM)
- STAR LC-24 24 tűs nyomtató
- külső drive

komoly Újság készítés:

- AMIGA 500 (3MB RAM)
- RGB monitor
- Min 20 MB winchester
- 14 MHz turbókártya
- HP laser II - lézernyomtató
- scanner
- szoftver: Pagestream 2.1, vagy Professional Page 2.0

- Második, illetve n-dik munkahely:

- AMIGA 500 (512K)
- RGB monitor
- szoftver: Transcript szövegszerkesztő

Áttekintés:

Mire való egy DTP program, mint a Pagestream?

Egy DTP program tulajdonképpen nyomdatermékek előállítására való. Szövegeket hasábockba tördelhetsz, különböző betűtipusokat használhatsz, és képeket is rakhatsz a dokumentumodba. Szórólapok, plakátok tervezésére is kitűnő, sokkal jobb, mint egy PRINTMASTER típusú program. Itt nincs rögzítve néhány fajta nyomtatvány (pl. levélpapír, hírdetés, stb.), hanem gyakorlatilag bármilyet elő tudsz állítani, ha megtanulod használni a programot.

PAGESTREAM

A programnak természetesen sok verziója van. A legújabb a 2.1-es, ami képes 24 bites színes képek beimportálására is, és dolgozik a compugraphic outline fontokkal is, amit a képernyőn is meg tud jeleníteni. Postscript type1 típusú fontokat küld minden printernek, akár egy 9tűsnek is. Nekünk az 1.8-as verzió áll rendelkezésünkre tesztelésre.

A Pagestream program egy WYSIWYG rendszerű DTP program, ami a "What you see is what you get", azaz "amit látsz, azt kapod" rövidítése. Amit tehát a képernyőn látsz, az lesz a nyomtatás eredménye. A kérdés általában az szokott lenni, hogy egy program mennyire WYSIWYG. Nos, a Pagestream ilyen szempontból (is) nagyon jó, bár a WYSIWYG azért itt sem 100%-ig igaz. Munkánk során a képernyőfontokat látjuk a képernyőn, amelyek egy bizonyos méret után "rűcskösek" lesznek, ami nem igaz a nyomtatásban megjelent vektorfontokra. Ha óriásira növeljük a betűk méretét, a képernyőn csak egy fekete téglalapot látunk a betűk helyett, de a nyomtatásban jó képet kapunk.



A program a betűket vektorokból építi fel, ami azért jó, mert nem meghatározott a betűméret, hanem tetszőlegesen állítható. Ha a betű egy adott méretű matrix-ban lenne megadva (bitmap-font), akkor ennek a nagyítása, illetve kicsinyítése során "rűcskössé", lépcsőssé válna, ugyanúgy, mint mikor a Deluxe Pant-ben nagyítasz fel egy betűt a képernyőn. Ennek a kiküszöbölésére egy megoldás, hogy a programhoz különböző méretben megrajzolt fontokat mellékelnek (pl. 9 pontos, 12 pontos, stb.). Ennek a módszernek a hátránya, hogy a betűk mérete nem lehet tetszőleges, ragaszkodnunk kell az adott méretekhez. Ennél fejlettebb megoldás a vektorfontok használata, ami azt jelenti, hogy a betűket vektorokból építik fel,

amelyek tetszőlegesen szorozhatók, tehát nagyíthatók bármilyen irányba. A pagestream ezt a megoldást használja, tehát a betűk tetszőlegesen nyújthatók függőlegesen, és vízszintesen is. Még a betű eredeti arányait sem kell megtartsuk, tetszőlegesen laposra, vagy éppen magasra torzított szövegeket készíthetünk. Ennek a módszernek azonban van hátránya is. Mivel az olcsó nyomtatók nem képesek közvetlenül postscript fontokat fogadni, a programnak először át kell alakítania ezeket a vektorokat egy pontmátrixá. Ez a folyamat sajnos elég lassú a standard A500-on, vagy 2000-esen. Ennek a számításnak a sebességét egy turbokártyával jelentősen meg lehet növelni. Drága ún. postscript nyomtatók azonban közvetlenül tudják fogadni a vektorfontokat, és maga a nyomtató hardverje végzi el a vektorok által leírt betű leképezését pontmátrixá - pillanatok alatt.

Nem beszéltünk még a képekről. Itt kétféle kell osztani a dolgot. A Pagestream tartalmaz egy csomó szerszámot vektoros grafikák készítéséhez, mint például vonalzó, körrajzoló, stb. Ezek segítségével egyszerűbb grafikákat a programon belül el tudunk készíteni, mint például grafikonokat, vagy elvi kapcsolási rajzot, stb. Direkt erre a célra készült azonban a sokkal többet tudó Professional Draw nevű program, amivel vektorokkal leírt rajzokat tudsz készíteni. Az így készült rajzoknak hatalmas előnye, hogy a méretüket tetszőlegesen lehet változtatni anélkül, hogy "rűcskössé" válna.

Lehetőségünk van továbbá IFF képek betöltésére is. Ezeknek a méretét is meg lehet változtatni, de akkor a kép minősége sokat romlik, hiszen ilyenkor nem vektorosan van tárolva. Lehetőség van IFF képek vektorossá alakítására is. Erre való a Vectortrace nevű program.

- folytatjuk -

SOHA ! ! !

Soha ne húzz ki, és ne dugj be semmit az RGB portra (a monitor csatlakozó), ha a gép be van kapcsolva !!! Tönkremehet az Ágnes, ami nem olcsó mulatság (kb. 1500 ATS) Ezt sajnos tapasztalatból mondom...

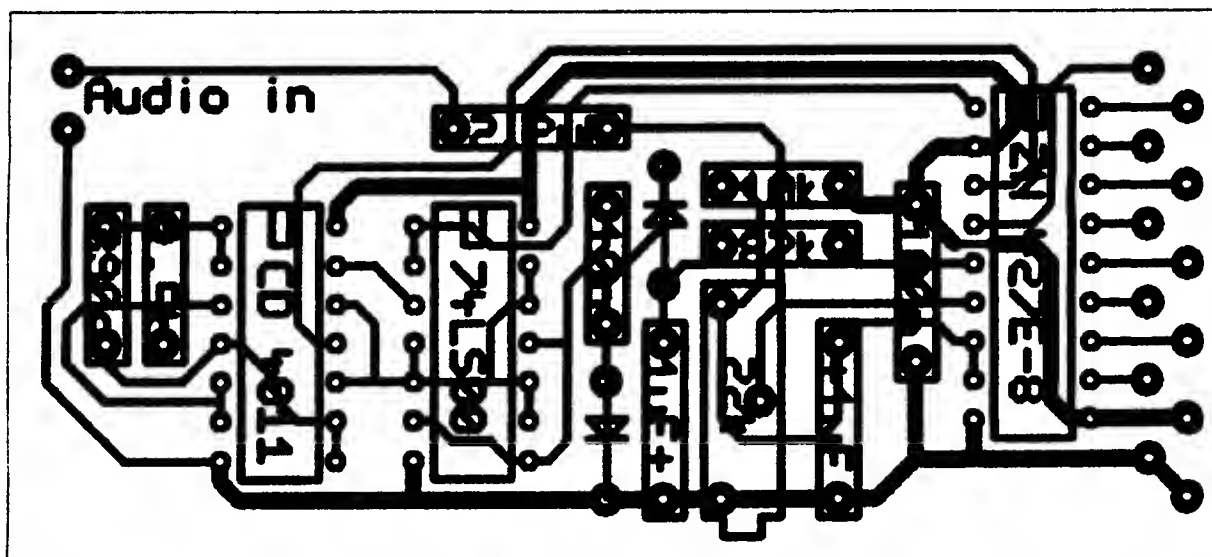
BB KING

Hardware

Rovatvezető: Compi

A nyák napja

Az AD áramkör működésének ismertetése után, ideje hogy a kapcsolás megépítésével, és élesztésével foglalkozzunk. Az áramkör eléggé bonyolult, kénytelen voltam a nyákot kétoldalasra tervezni. Mivel kis hazánkban szinte lehetetlen furatgalvanizált panelt készíttetni elfogadható áron, az átkötések számát igyekeztem a lehető legkisebbre csökkenteni. A közölt fóliarajzok közvetlen fotózását nem ajánlom, mivel a rajzolat finomsága ehhez nem elég. Ajánlatos tehát előbb klisé készíteni, és a nyákot arról fotózni. A panelra csatlakozó vezetékek számára nem terveztem furatokat, mivel tapasztalatom szerint a közvetlen felületre forrasztás kényelmesebb. A beültetési rajzon számmal jelölt csatlakozóhelyeket az Amiga parallel portjának megegyező számú lábaihoz kell kötnünk. (Természetesen csatlakozó használatával.) A GND–vel és a +5V–tal jelölt helyekre kell beadni az áramkör +5V–os tápfeszültségét. Ezt levehetjük a külső floppy csatlakozójáról is, de érdemesebb az áramkörhöz egy külön tápot gyártani, természetesen stabilizáltat.



Élesztgetés

Bevezetésként figyelmeztetnem kell mindenkit arra, hogy az áramkört ne próbáljuk szájból–szájba lélegeztetéssel élesztetni, mert egyrészt úgysem használ, másrészt még jó kis áramütést is kaphatunk.

Legjobb, ha az élesztést részegységenként végezzük. Ehhez először is az kell, hogy minden IC–t foglalatba rakjunk, hogy az élesztés folyamán egyenként tehessük be őket a helyükre, természetesen megfelelő helyzetben. Ha valamelyik IC–t véletlenül fordítva rakjuk be, szemet és fület gyönyörködtető fény és hangjelenségben lehet részünk. Az élesztést az összes passzív alkatrész beforrasztása után a segéd tápfeszültség–generátorral kezdjük. Itt mondom el, hogy minden IC–t természetesen kikapcsolt állapotban szabad csak a helyére tenni. Első lépésként az üres IC foglalatokon ellenőrizzük, hogy megvan–e a +5V–os tápfeszültség. Ha ez rendben van, rakjuk be a diódák mellett lévő CD 4011–et a helyére, és újra kapcsoljuk be az áramkört. Ekkor a ZN 427–es foglalatának ötös lábán –4, –5 voltot kell mérnünk. Most az óragenerátor élesztése következik. Ehhez tegyük a foglalatba a másik CD 4011–et. Ekkor a ZN hármas lábán meg kell jelenjen egy kb. 100 kHz–es négyszögjel. Most helyükre tehetjük a többi IC–t is, és csatlakoztathatjuk az áramkört a géphez. Hívjunk be egy digitalizáló programot (pl. AudioMaster), és a sampler részénél válasszuk a MONITOR funkciót. Ezután mindkét trimmerpotit tekerjük középpállásba. Ennek hatására a képernyőn látható csík kb az ablak közepére kell álljon. Miután a helikális potméterrel a AD nyugalmi szintjét pontosan középre állítottuk, már csak jelet kell adnunk a BE felíráttal jelzett pontokra mondjuk egy magnóból, és áramkörünk máris működésre kész.

A hangdigitalizáló megépítéséhez és használatához sok sikert kívánok !

Megjelent az első magyar nyelvű amigás
lemezújság:

A
GURU

Tartalmából:

aktualitások

játék és felhasználói programleírások

grafika és zene pályázat

nyílt levelezés rovat

Lemezen megrendelhető:

(lemezen küldjük)

1399 Budapest, Pf. 701/GURU

Ára:150. – Ft

Íme egy példa arra, hogy milyen hirdetéseket közöl le az AM abszolút ingyen, közérdekű információ
gyanánt:

(Trararaaaraaaa !)

ÖRVENDJETEK AMIGÁSOK !

Az Örs Vezér téren, a Metró melletti nagy zöld épületben levő
C64–Amiga klub tárt karokkal vár benneteket !

(éljen, éljen, éljen)

Pontos cím: Budapest XV.ker Fehér út 1. VII.emelet. Nyitva
minden kedden 16–20 óráig.

Belépő nincs! Kilépő sem !

A KLUBBAN a következő gépekkel foglalkozunk:

Spectrum, C16, C+4, C64, Amiga

Rendszeresen kaphatók lemezek 5.25 –350Ft/doboz, 3.5 –800
Ft/doboz, valamint üdítők !

A klub 1989 novemberében alakult, és azóta folyamatosan bővül
a taglétszáma. Jelenleg 70 tagunk van + aki előfordul.

A klubvezető neve: Szelepcsényi Ferenc.

MI ÚJSÁG ?

2 Milliő Amiga

A Commodore cég nemrég jelentette be hivatalosan, hogy az eladásai elérték a 2 milliő gépet az Amiga családban. Számolgassunk egy kicsit, becsüljük meg, hogy ez mit jelent!

Ez azt jelenti, hogy a Commodore, és a céget körbevevő dealerhálózat 2 billió dollárt vételezett be az elmúlt néhány év alatt. A két milliő gép azt jelenti, hogy minden 2500-dik embernek a világban van egy Amigája. Ha sorba raknád őket kb. 800 Km hosszú Amiga "járdád" lenne. Ha egymás tetejére tennéd a "fiúkat" egy tíz emeletes piramist alkotnának, aminek az alapja akora lenne mint egy futball-pálya. A súlyuk kb. 10.000 tonna lenne, aminek a szállításához 40 db B52-es bombázóra lenne szükség. 200 Megawatt áramra lenne szükség, hogy mindet futtatni tudj az összeset a hozzátartozó perifériákkal együtt. A gépeknek összesen 2000 gigabyte memóriája van. Ez már valószínűleg elé lenne, ha minden A500-as mamaboardján ennyi lenne...

Tovább folytatva a fantáziálgatást: ennyi Amiga 8000 db Cray2-es szuperszámítógép számítási teljesítményével egyenértékű... Azt hiszem így már nem kellene sokat várnunk egy Sculpt képre... Ha az összes monitort egymáshoz tapasztanánk, egy 800 méter széles, és 570 méter magas óriási jelzőt kapnánk. Ha már világviszonylatban ilyen nagyot fantáziáltunk, gondolkozzunk el azon, hogy hány Amiga lehet Magyarországon! Ha minden ember a Földön egyenlő mértékben részesülne a gépekből akkor Magyarországon 800-nak kellene lennie. Ennél pontosabb becslést ezekből nehéz kihozni. Feltételezhető, hogy Afrikában, Kínában sokkal

kisebb az egy főre jutó Amigák száma míg nálunk, ugyanakkor az USA-ban és Germániában ez a hányados szinte egy! Az arányokon nem érdemes filozofálgatni. Szerintünk realisztikusnak egy kb. 4000 Amiga tulajdonos tekinthető. Gondoljunk bele abba is, hogy abba 2 milliőba beletartoznak azok az 1000-esek, és 500-asok tömkelege is, amit a gyámak visszavettek, hogy a gépüket 2000-esre cseréljék, számolnunk kell azzal is, hogy cégek ezreinél van több, mint tizenöt gép is, stb.

Nyugati könyvek forintért ?

Nem, sajnos nem az történt, hogy államunk totálisan impotens kormánya végre konvertibilissé tette volna a forintot, vagy bármit tett volna az ilyen típusú vállalkozások könnyítésére, (sőt Bod Péter Ákostól olyan megnyilvánulást hallottam a napokban, ami még a legelvakultabb marxista közgazdásznak is becsületére vált volna...). Nem, nyugatról árut importálni még mindig ugyan olyan bonyolult, és a lehető legjobban igyekszik a kormányzat, hogy az ilyesmivel foglalkozókat tönkre tegye idegileg és anyagilag.

A hír tulajdonképpen csak annyi, hogy a lapunk hátoldalán reklámozó Typotex KFT.-hez útou van az első kliensek által megrendelt könyvszállítvány. Reméljük mire ezt olvasod, már meg is érkeztek ezek a könyvek. Így a szállítási határidő kb. egy hónap, ami a jövőben valószínűleg akár két hétre is lerövidíthető lesz majd. Így ha szükséged van valamilyen igazán színvonalas szakirodalomra nálunk megrendelheted, és

két héten belül már a kezeden is van.

STORMBRINGER H530

Szokásos hardverűségeink közül az egyik figyelemreméltó egyed ez a kártya. Az Intelligent Memory nevű cég, amely jó hírnévre tett szert a kitűnő minőségű kiegészítők gyártásában ebben az évben hozta ki "hurrikán" továbbfejlesztését, a "viharhozót". Aki rendszeresen olvassa az AM-et az esetleg emlékszik az előző számban elkövetett álmódosításunkra az 500-asba építhető Arriba hardiskkel kapcsolatban. Nos, azt hiszem ha egy A500-ban egy Arriba, és STORMBRINGER H530-as lenne, és egy kiállításon kiraknád a hónod alól a mit sem sejtő közönségnek, lenne meglepetés! Egy ilyen gép többször gyorsabb lenne, mint a leggyorsabb 3000-es.

A H530-ban egy 68030-as CPU van, ami 16, 28, 36, 54 MHz-en futtat, valamint egy 68882 FPU (lebegőpontos aritmetikai koprocesszor), ami 16,28,36,60 MHz-en. A kártyán továbbá 8MB szupergyors 32 bites RAM-nak van hely, ami lépésekben beülthető és teljesen autokonfigurációs. A kártya minimális kiépítésben 2195 DM-be kerül, amit egyelőre sajnos nem tudunk, hogy mit tartalmaz. (Hány MHz, mennyi RAM)

Ha már itt tartunk leírjuk a cég másik két kártyáját is. A HURRICANE 500-as 16 MHz-es 68020-as procin alapul. Van rajta hely a 68882-es FPU-nak (16,28,36 MHz), és 4MB 32 bites nulla várakozási státusz RAM-nak. Ez a kártya a 16MHz-es CPU-val, és 1MB RAM-mal 1095 DM-be kerül.

Hogy a 2000-esről is ejtsünk néhány szót: A HURRICANE 2000

kártya egy kicsit zűrsnek tűnik a hirdetés alapján. Van rajta egy SCSI kontroller, de nem írják, hogy 8, illetve 16 bites, ami bedig egyáltalán nem mindegy a sebesség szempontjából (az SCSI kontroller harddisk és egyéb ún. SCSI perifériák illesztésére való). Arról sem szól a fáma, hogy van-e rajta FPU, vagy egyáltalán hely neki a kártyán. 4MB 32 bites RAM bővítési lehetőség és direk elérés a 16 bites RAM kártyákhoz. Az igazat megvallva én ha a 2000-esemhez turbókártyát vennék, (feltéve ha...) a GVP3001-es mellett döntenék. Hogy miért arról majd később, ha egy teljes cikket a turbókártyáknak szentelünk.

H2800 28MHz Kit DM 3495.00

H2800 50MHz Kit DM 5995.00

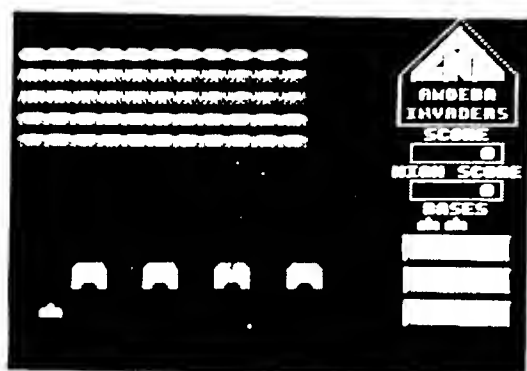
Intelligent Memory GmbH
Adam Opel Str. 10
6000
Frankfurt 60
Tel.: 00-49-69-41 0071-73

Amiga számítógépek
és
megbízható
javítás és
alkalmazási
tanácsok
(és a boot
kapcsoló
chip-faszt
kapcsoló, stb.)
és a tapasztalat
tanácsok
és a tapasztalat
tanácsok
és a tapasztalat
tanácsok
és a tapasztalat
tanácsok

ON DISK 03:

**Az úságban található listák
"CAR" - ray-tracing animáció
Public Domain játék:**

Galaxy



ON DISK 01, 02:

SUPERHAM KÉPEK (Bohóc, rózsza)

(A Super-Ham egy olyan képformátum, ahol soronként új palettát használ a gép. A Minőség ? Fantasztikus !)

AM Ray-tracing animáció

TRON Basic játék !

Clickdos

Soundtracker lejátszórutinok



PIAC

Az MNB árfolyamai:

USD - 69.12 Ft

DM - 47.32 Ft

ATS - 6.69 Ft

GBP - 137.29 Ft

Február 11.-i keresztárfolyamok:

	USD	DM	ATS	GBP
USD	1	1.45	10.95	0.50
DM	0.68	1	7.03	0.34
100 ATS	9.77	14.223	1	4.91
GBP	1.99	2.89	20.37	1

A Novotrade 2C áruház érvényben lévő árlistája:

C-64 alapgép	19.400.Ft
VC 1541/II. drive	19.900.Ft
1530 datasette	3.500.Ft
C-64 mouse	3.200.Ft
MPS 1230 nyomtató	24.900.Ft
Monochrome monitor (12" zöld)	14.000.Ft
Joystick	1.300.Ft
Amiga 500 alapgép	59.900.Ft
Amiga RF modulátor	3.290.Ft
Amiga tárbővítő (1-MB)	12.900.Ft
Amiga tárbővítő (1-MB órás)	14.500.Ft
Amiga hard disk (20-MB)	54.500.Ft
Color stereo monitor 1084 S. (14")	35.000.Ft

M.A.S.T. Have it !

Josefweg 45
A-8043 GRAZ
Tel.: 0316-373763
Fax.: 0316-383382

Colorburst 24 bites kártya: 7099 ATS

STARBLACER turbokártya:

- ALAP (0MB): 1899 ATS, (2MB): 2899 ATS
 - SCSI kontroller: 8 bites: 549 ATS 16 bites: 1099 ATS
 - 14 MHz 68000 999 ATS 16 MHz /030: 4699 ATS 25MHz/030: 6299 ATS
- Brainstorm Atari ST emulator: 759 ATS !!!**

Börze

AZ Amiga Magazin ingyenes hirdetési rovata. Extra hirdetés: 100 Ft

Lépj más dimenziókba!
KICKSTART V.2.0
(lemezen), Workbench
V2.0, Extras V2.0. A
három együtt 1000 FT
vagy 15 db új lemez
(3.5"-os)
Veszprém, pf. 296. 8200

Original No-Name 3.5"
DSDD mágneslemez eladó
650 Ft/doboz áron amíg a
készlet tart. Érdeklődni:
Gerencsér András
1667-305, az esti órákban.

ACTION REPLAY
cartridge Amigához, és
C-64-hez! Egyéb C-64-es
cartridgeok és tartozékaik
is eladók. Tel.: 1645-442

Amiga 500 és C64
programokat cserélek és
eladók. Listát kérek és
küldök. Főleg felhasználói
és játékprogramok
érdekelnek.
Címem: Németh András,
9081 Győrújbarát, Veres
P.u.23.

3.5"-os DSDD lemezek
eladók. Ár: 700 Ft/10 db,
programokkal együtt 900
Ft/10 db. Kérésre
programlistát küldök.
Érdeklődni lehet 18 óra
után: 155-1688

PHILIPS monitor eladó.
Irányár 25.000 Ft.
Pető Zoltán, Paks, Ifjúság
u.10 7030.
Tel.: (75) 10-602

FELHASZNÁLÓI és
szimulátor programok
cseréje. Széchenyi János
2701 Cegléd, Pf.55/HK.

C1581-ES DRIVE-omat
elcserélném 3.5"-os
lemezekre, vagy 3.5"-os,
5.25"-os drive-ra
Amigához. Eladó Amiga
500 (1MB RAM)
Vasics Tamás
8855 Belezna, Kossuth u.
40.

A FLIGHT SIMULATOR
magyar nyelvű leírását
keresem megvételre
Farkas László
5435 Martfű. Szolnoki út
13.

ELADÓ: A500 + 1MB +
Thomson Color Monitor,
110 db lemez, lemeztartó,
Jumbo-joy, egér.
Tel.: (83) 11-626

AMIGA PRG.-OK
(25Ft-ért) eladók.
KERESEM az Imagine,
Obitus, Lemmings című
programokat. Turbosilver
és Assembler tanfolyam
magyar nyelvű leírása eladó
700/700 Ft.
Kotróczó Balázs
1124 Bp., Hegyalja út 63
Tel.: 165-5110

DEMÓK,
FELHASZNÁLÓI
programok cseréje 1MB
Amigán! Érdeklődni
levélben!
Bácsi Péter
1134 Bp., Gidófalvy L.u.9
VIII.em.3

ELADÓ: RED SECTOR
DEMOMAKER, akár az
eredeti is, német leírással.
Keresek mindenféle olyan
programot, amivel valamit
készíteni lehet.
Krajczár Róbert
3700 Kazinbarcika, Kun
Béla u.13.

Képek scannelését
(digitalizálását) vállalom.
További info-ért írj a
következő címre:
Kéri László
1031 Bp., Amfiteátrum
u.26.

AMIGA PROGRAMOK
20FT/db és Noname
lemezek 95 Ft/db eladók.
Kérésre listát küldök.
Lajcs Róbert
6723 Szeged Szilléri SGT.
24/A II/6

PROGRAMCSERE Amiga
500-ra! Kérésre listát
küldök!
Kersztalvi János
1034 Bp., Doberdó út 4.

Magyar nyelvű
szakkönyveket vagy
fénymásolatokat vennék
A500-hoz.
Kovács Gábor
2030 Érd, Sárd u.5

AMIGA 500-hoz
hangdigitalizáló, 0.5-2MB
közötti memóriabővítő,
3.5" lemez eladó.

Szirovica Ernő
6771 Szőreg Szerb u.30
Tel.: (62) 55-061

EXTRA AMIGA
PROGRAMOK kaphatók
25 ft/disk. Amiga lemezek
eladók 950Ft/10db
Magyar nyelvű Amiga
szakkönyvek kaphatók.
BASIC, DOS, gépi kódú
felhasználói kézikönyv.
Haár László
1133 Bp., Dráva u.11
Tel.: 1732-008

KERESEK C64-re olcsó
nyomtatót. Ajánlatokat a
következő címre kérem:
Mihály Győző
3441 Mezőkeresztes,
Rákóczi út 34/1

ELADÓ 3.5" kölső drive!
(Amiga)
Dara Viktor
5100 Jászberény, Fáy
A.u.21
Tel.: (57)12-469

C64 + floppy + magnó +
80 lemez tartóval + 2 joy
eladó.
Rácz Tamás
3348 Szilvásvárad, Park
u.19
Tel: (36) 55-245

AMIGA 500-hoz RAM
bővítőkártya eladó.
Tel.: (62) 84-181

SWAPPERT COOL
CONTACTOKKAL,
Angol nyelvtudással, hot
stuffokkal, NEM
játékorientáltsággal
keresünk.
Nexus 6 of 23 celsius crew
8200 Veszprém, pf.296

AMIGA PROGRAMOK
nagy választékban eladók
lemezzel együtt (80 ft-db),
válaszborítékért listát
küldök.
Amiga shop!
1213 Bp., Határ u.103

EZ FANTASZTIKUS!
Millió adat, cím,
telefonszám. Könyvtárak,
video, és hangkasszéták
nyilvántartására,
visszakérésére alkalmas
a DATAMAT PLUS
AMIGA! Magyar nyelvű
könyvvel!
Telefon: 185-3642

AMIGA 500-ra keresek
olcsón játék és felhasználói
programokat. Keresem a
Last Ninja II, és a Day of
Thunder nevű
programokat.
Nagy Péter
3881 Abaújszántó, Petőfi
út 32

IF YOU HAVE ANY new
idea, or you are a cool guy
then write to NEXUS 6 of
23 Celsius Crew. Grafix,
Demo-user swap.
Veszprém, 8200 Pf.296.
Bye!

AMIGA PROGRAMOK
20 Ft-ért. Csere is
lehetséges. C64-hez
nyomtató eladó!
Vadon Zoltán
Telefon: 1669-539
(délután)

AMIGÁRA programokat
cserélnék, keresek
hardware leírásokat
(genlock, képdigitalizáló,
modem). A címem:
Sipos István
2013 Pomáz, Orgona út 57.

ELADÓ! Márkás (3M,
BASF...) 5.25" lemezek
kedvező áron!
Agócs Péter
4211 Ebes, Gázláng u.1.
Tel.: (52) 66-352

ÉPÍTÉSZETI ÉS
GRAFIKUS
SZOFTVEREKET
KERESEK. AZ E
TÉMÁVAL
FOGLALKOZÓKKAL
LEVELEZNÉK.
Bogár László
8734 Somogyiszita, Ady
Endre u.8.

**AMIGÁSOK! Szuper
programokat olcsón,
gyorsan? Érdekel a
computer zene? Írj!
válaszborítékért részletes
tájékoztatót és listát
küldök! Programjaimról
részletes felvilágosítást is
tudok mellékelni!
Constructions Kit-eket
(Shoot em up, stb.)
keresek. AMOS leírás
érdekel, írj!
Dervallás Tibor (Cranon)
8256 Ábrámhegy
Patak u.6.**

MI LESZ ?

**VIDEOLUX
professional
ismertető**

**Varga Csaba:
hideg reset
kapcsoló**

**Deluxe Paint...
Pagestream...**

**Teszt:
Digiview Gold
4.0**

**Amiga a
Disco-computer**

**És még egy
csomó más...**

AMIGA MAGAZIN

Csak az Amigáról fehéren-feketén

Az AM postacíme:
H1075 Budapest, Tanács krt. 5/C.
(Csak postai küldemények fogadására.)

Felelős kiadó: HORIZON-multiplan KKT.

Szerkesztők: Bordás Bence és Ürmössy Zoltán

Az februári szám munkatársai:

Bódy Attila
Beleznai Annamária
Ürmössy Zoltán
Jászberényi Márk
Kiss Sándor
Kosir Attila
Kéri László

Megjelenik havonta, ára 148.- Ft.
Megrendelhető az AM postacímén a lapban
található megrendelőlapra vagy levélben.
Előfizetési díj:
negyedévre: 444.- Ft
fél évre: 888.- Ft
egy évre: 1768.- Ft

Nyomda:
Alfaprint Nyomdaipari Szövetkezet

A szerkesztés a Horizon-multiplan kkt DTP
rendszerén készült.

Na látod! Ugye mondtuk, hogy amint lehet, olcsóbbak leszünk!

Ne felejtsd el visszaküldeni a visszacsatolást, amilyen hamar csak tudod!
Használd ki az ingyenes hirdetési lehetőséget, válj meg a fölösleges belső bővítődtől, és bővítesd ki a géped alaplapon (csak ha 1.3-as géped van)! Így az 1.2-esek olcsón jutnak belső bővítőhöz, de azért te is jól járhatsz. Szerezz Amiga Amigókat, levelező, cserepartnereket!

Rendeld meg az AM-et! Ugye tudod, hogy az egyetlen biztos módja az újságunk megkapásának, ha tőlünk megrendeled. Ez azt jelenti, hogy a Visszacsatoláson kitöltöd az adataidat, és a megrendeléssel kapcsolatos egyéb adatokat (milyen géped van, mi érdekel, stb.) Ezt add postára, és add föl az előfizetési díjat rózsaszín postautalványon (a postán kell kérni) a postacímünkre:

AMIGA MAGAZIN

1075 Budapest

Tanács Krt.5 c

Az előfizetési díj:

Februári szám: 148 Ft

Negyedévre: 444 Ft

Fél évre: 888 Ft

Egy évre: 1776 Ft

Aki már megrendelte a 190 forintos újságot, az így hosszabb ideig fogja kapni.

Aki egy évre fizette be, az plusz 4 hónapig fogja kapni.

Aki fél évre, az plusz két számot kap újabb befizetés nélkül.

Aki negyedévre fizette be, annak majd kevesebbet kell fizetnie, hogy meghosszabítsa a megrendelését.

SZÁMÍTÁSTECHNIKAI KÖNYVIMPORT FORINTÉRT!
Gyors lebonyolítással! Az árfolyam + 30%-os áron

MEGRENDELHETŐ KIADVÁNYOK:

Duncan: Advanced OS/2 Programmin , Microsoft Press (782 old.)	2150 Ft
Viescas: Guide to SQL (The Standard Relational Database Language), Microsoft Press (102 old.)	640 Ft
Postscript Language : Refreence Manual, Addison Wesley	1830 Ft

AMIGA szakirodalom:

Sawusch-Prochnow: Things to Do With Your Amiga , TAB (195 old.)	10 GBP
Commodore Business Machines Inc.: Amiga Hardver Reference Manual , Addison Wesley (325 old.)	24.95 GBP
Halfhill-Brannon: Advanced Amiga Basic , Compute! (466 old.)	16.95 GBP
Commander: Amiga Assembly Language Programming , J.Wiley (kb. 250 old.)	10.80 GBP
Rugheimer-Spanik: Amiga Basic Inside & Out , Abacus (547 old.)	18.95 GBP
Bleek-Maelger-Weltnem: Amiga Tricks & Tips , Abacus (315 old.)	14.95 GBP
Leemon: Inside Amiga Graphics , Compute! (308 old.)	16.95 GBP
Bleek-Langlotz: Das große AMIGA-500 Buch , Data Becker (528 old.)	49 DM
Polk: Die besten AMIGA Utilities , Data Becker (403 old.)	39 DM
Langlotz-Vignjevic: Das große DPaint-III-Buch , Data Becker (393 old.)	39 DM

Bővebb felvilágosítás munkanapokon 9-16 óráig, Lesnyik Lászlónénál (115-2337)

Várjuk megrendeléseiket!

TypoTeX Kft. Elektronikus Kiadó
1015 Budapest, Batthyány u. 14.; Tel.: 201-3317; Fax: 162-1804